VISUAL ARCTIC NAVIGATION: TECHNIQUES FOR AUTONOMOUS AGENTS IN GLACIAL ENVIRONMENTS

A Thesis
Presented to
The Academic Faculty

by

Stephen Williams

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2011

VISUAL ARCTIC NAVIGATION: TECHNIQUES FOR AUTONOMOUS AGENTS IN GLACIAL ENVIRONMENTS

Approved by:

Professor Patricio Vela,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Edward J. Coyle
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ayanna M. Howard, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Henrik Christensen
College of Computing
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: June 13, 2011

DEDICATION

For Joy and David Williams,

and Kimberly Rees, without whom life would be much less interesting.

ACKNOWLEDGEMENTS

This work was supported by the National Aeronautics and Space Administration under the Earth Science and Technology Office, Applied Information Systems Technology Program. I would also like to express my gratitude to my collaborators Dr. Derrick Lampkin, Pennsylvania State University, for providing the scientific motivation for this research, Dr. Magnus Egerstedt, Georgia Institute of Technology, for providing his experience in multi-agent formations, and Dr. Matt Heavner, Associate Professor of Physics, University of Alaska Southeast, for providing his expertise in glacial field work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Arctic regions are thought to be more sensitive to climate change fluctuations, making weather data from these regions more valuable for climate modeling. Scientists have expressed an interest in deploying a robotic sensor network in these areas, minimizing the exposure of human researchers to the harsh environment, while allowing dense, targeted data collection to commence. For any such robotic system to be successful, a certain set of base navigational functionality must be developed. Further, these navigational algorithms must rely on the types of low-cost sensors that would be viable for use in a multi-agent system. A set of vision-based processing techniques have been proposed, which augment current robotic technologies for use in glacial terrains. Specifically, algorithms for estimating terrain traversability, robot localization, and terrain reconstruction have been developed which use data collected exclusively from a single camera and other low-cost robotic sensors. For traversability assessment, a custom algorithm was developed that uses local scale surface texture to estimate the terrain slope. Additionally, a horizon line estimation system has been proposed that is capable of coping with low-contrast, ambiguous horizons. For localization, a monocular simultaneous localization and mapping (SLAM) filter has been fused with consumer-grade GPS measurements to produce full robot pose estimates that do not drift over long traverses. Finally, a terrain reconstruction methodology has been proposed that uses a Gaussian process framework to incorporate sparse SLAM landmarks with dense slope estimates to produce a single, consistent terrain model. These algorithms have been tested within a custom glacial terrain computer simulation and against multiple data sets acquired during glacial field trials. The results

of these tests indicate that vision is a viable sensing modality for autonomous glacial robotics, despite the obvious challenges presented by low-contrast glacial scenery. The findings of this work are discussed within the context of the larger arctic sensor network project, and a direction for future work is recommended.

CHAPTER I

INTRODUCTION

In the last decade climate change has become a household term. Yet, scientists still lack important data needed to accurately model and predict climate effects [38]. In particular, it is believed that arctic zones are more sensitive to the effects of climate change, making weather-related observations from these regions of particular importance [18, 102].

Automatic Weather Station (AWS) networks are popular tools for *in situ* measurements. These instruments remain fixed in a single location and are usually equipped with several weather-oriented sensors, such as pressure sensors, anemometers, and pyranometers [111]. Given the immobility of these devices, the accuracy of measurements taken becomes a function of sampling and estimation capacity relative to the entire network. Each AWS unit spans a limited radius of coverage and scientists must consider other units in the network, relying more heavily on extrapolation methods to obtain a breadth of coverage in an area. For example, the Greenland Climate Network (GC-Net) averages only one station per 100,000 km^2 [110], while the Antarctic Meteorological Research Center (AMRC) and Automatic Weather Station (AWS) program provides even sparser data [65]. Any data collected from one unit represents a single point on a map, useful only as a heuristic to indicate what changes may be taking place [23].

Satellite-based instruments are now routinely used to monitor weather conditions. For example, NASA's Earth Observing System (EOS) provides a potential wealth of information regarding the state of the environment through a variety of on-orbit sensing capabilities. NASA's Ice, Cloud, and land Elevation (ICESat) satellite [142] is

mapping the earth's surface using a Geoscience Laser Altimeter System (GLAS), while Landsat is capturing high resolution imagery. Several climate-oriented instruments onboard the Terra [84] gather such information as the Land Surface Temperature (LST) and pollution levels. However, orbital measurements of ground-based quantities are heavily influenced by external factors, such as air moisture content or surface emissivity [142]. Detailed atmospheric models are created to compensate for these factors, but the satellite system must still be validated to ensure the accuracy of the instrumentation and correction model [108, 125]. For proper on-orbit sensor validation, calibration sites should be selected to cover the expected range of global surface properties. Further, data should be collected at a variety of scales, similar in size to the single pixel area of the produced data product [84]. Calibrating over areas that closely represent the measurement areas of interest enhance the accuracy of the model. Validation and correction are particularly important for arctic environments, as the unique surface properties of packed snow and ice are poorly modeled by any other terrain type.

Human-led field campaigns provide the highest resolution for these types of weather measurement data. GPS and Ground Penetrating Radar (GPR) surveys are useful calibration techniques for validation of remote sensing equipment such as GLAS [108]. These surveys require constantly manned equipment with integrated sensing, and carefully planned navigation paths. Though the coverage area is considerable for *in situ* trials (100 km^2), the duration of these field experiments is potentially more strenuous on the scientists performing the tests. An example of this methodology has been used to validate the LST recorded by Moderate Resolution Imaging Spectroradiometer (MODIS) on-board the Terra satellite [14]. A 1 km^2 region of a large rice field in Spain was selected as the validation site. The rice field offered a large, flat area that was uniformly covered in vegetation. Hand-held temperature sensors were stationed at several points within the test site. During the satellite overpass

event, GPS-registered temperature readings were collected and logged several times a minute as the sensor was moved over a 100 m traverse. Due to the satellite orbit, only a handful of overpass events occur within the validation site during each repeat cycle. Further, as the satellite repeat cycle is 183 days long, only one such cycle occurs each year within the growing season, during which the site has uniform vegetation coverage. During the three year period over which these experiments were performed, only 11 validation events of MODIS were recorded. In order for these on-orbit sensors to be thoroughly validated on all types of terrain, a more efficient and cost-effective mechanism is needed to perform these *in situ* data collection campaigns in remote regions.

Field robotics has a distinguished history of collecting scientific data from the far reaches of not only this planet, but the solar system. Perhaps the most famous example is the Mars rovers, Spirit and Opportunity [109], which have examined Martian geological history, or the recent Phoenix lander, which confirmed the existence of water ice on Mars [105]. Closer to home, robotics systems have taken atmospheric samples from the mouths of volcanoes [86], sifted through rubble at disaster sites [85], and explored the deepest reaches of the ocean [139]. In all cases, robotic systems have been deployed to gain insight into areas that are too remote or too dangerous for humans, adding to our collective scientific knowledge. Instead of relying on human-led campaigns, a mobile robotic sensor network has been proposed that greatly mitigates the human resource requirements [133, 135].

The proposed system centers on decentralized autonomous robotic nodes outfitted with a scientific sensor package for the collection of ground-based environmental measurements. The reconfigurable network of robotic nodes can be tasked to collect data at a location and spatial resolution that is optimal for the scientific needs. Scientists would upload a target location and desired sensor resolution to the robotic team, which would then be in charge of executing the task [135]. Due to the typical

distances between existing weather stations, it could take tens or hundreds of hours for the network to reach its final goal, making teleoperation an impractical solution. Instead, each robotic node will be responsible for navigating from the base location to the designated target autonomously, adapting navigation plans in response to the perceived conditions of the environment. Further, as many rovers will be required to create a usable sensor network, each unit must be inexpensive. This pushes the design away from centimeter accuracy GPS and military-grade IMU sensors and towards consumer-grade sensing technologies.

In particular, vision is an attractive option. It is the sensing modality relied upon most by humans, and it has been shown effective for both the Mars rovers [47] and DARPA Grand Challenge vehicles [80]. Compared with laser scanners, even high resolution cameras are light, low power, and inexpensive. If vision is to be utilized as the primary sensing modality for the navigation system, it must be able to accommodate the tasks of obstacle detection, self localization, and local area mapping.

The area of visual obstacle detection has been studied exhaustively by the robotics community, but the solutions are generally specific to the environment. The DARPA Grand Challenge and Mars rovers prompted development in desert processing techniques, while the Urban Grand Challenge pushed research forward in structured, city environments. However, glacial regions present unique hazards that are visually distinct from the rocky terrain of Mars or the structured environments of urban cities. These hazards are predominantly "slope based," consisting of inclines and gentle perturbations in the glacier surface caused by changes in the underlying rockbed, as well as more dramatic slope changes in the form of surface cracks, lake basins, and protruding mountain peaks. Since any deployed robotic system will have a limited range of slopes that it is capable of traversing safely, new image processing techniques have been developed that are capable of characterizing the terrain slope, and hence the traversability of the terrain. The details of the slope estimate technique are described

in Chapter 3.2.

However, before any meaningful analysis of terrain obstacles can be performed, the foreground region must first be segmented from the acquired image. This serves to focus subsequent processing on a smaller, targeted region of interest. Typical image segmentation algorithms use information local to the examined pixel to make segmentation decisions. However, the properties of glacial images make local examination problematic. Overcast skies, common in glacial environments, often share the same color range as the ground plane snow. Further complicating segmentation, the clouds and ground plane often intersect visually, making the determination of the horizon difficult. When analyzing these images, humans tend to scan the image for visual cues in the form of strong horizon line segments. These line segments are then extended into image regions where the horizon is more ambiguous. Using this type of strategy, a ground segmentation method has been devised in Chapter 3.1 that is capable of correct operation in these challenging situations.

To perform localization from visual input, simultaneous localization and mapping (SLAM) algorithms are generally employed. For the proper operation of SLAM, feature points within the image stream must be reliably extracted and matched. However, glacial environments generally lack these types of distinctive features. To that end, a set of image preprocessing steps have been developed to boost the feature extraction performance, and a study of the detection performance of common feature detection algorithms has been conducted in Chapter 4.2.

Even with a source of reliable image features, SLAM systems are incremental in nature. Each update is based on the prior position estimate. As the system runs, small errors accumulate resulting in significant localization error over time. In order to remove this drift, global position information in the form of low-accuracy GPS data has been fused with vision-based SLAM. The method of this integration, along with additional implementation details of the visual SLAM system are presented in

Chapter 4.3.

Finally, as the number of impassible obstacles in the environment increases, the need for global path planning becomes acute. While the algorithms employed by global path planners differ significantly, all planning strategies require a map on which to plan. While the visual SLAM system produces a set of 3D point estimates that lie on the terrain surface as a byproduct of the localization process, this type of map is not readily utilized by common path planning algorithms. Instead, a terrain reconstruction method is presented in Chapter 5 that creates a topographic terrain map. This method combines the sparse landmark position estimates from the localization system and the dense slope estimates from the terrain assessment using a statistical construct known as a Gaussian process (GP). It is shown that the GP framework is capable of generating a reasonable terrain model using only the sparse landmark points, outperforming the standard triangular mesh interpolation, particularly at large distances. Further, it is shown that the incorporation of slope information into the GP significantly improves the reconstruction, something not easily integrated into simpler interpolation schemes.

The remainder of this chapter introduces background information related to the deployment of robotic systems in the glacial terrain. First, a general description of glacial terrain is presented, including a discussion of the types of hazards likely to be encountered. Next, previous efforts in deploying glacial robotic systems are detailed, and their respective navigation systems outlined. Finally, as glacial-specific navigation systems have not been developed, a general overview of the current state of field robotic navigation systems is provided, with special emphasis on vision-based methods for localization, obstacle detection, and map building.

1.1 Environment

Glacial ice sheets, the largest glacial formations, cover much of Antarctica and Greenland [48]. These ice sheets are dome-shaped, contained within a basin formed by the underlying mountain range system. Ice fields, such as the Juneau Ice Field in Alaska [57], are smaller and thinner than glacial ice sheets, typically covering several thousand square kilometers. Due to the thinner nature of ice fields, terrain changes of the underlying rock bed are transmitted to the ice surface, resulting in undulating terrain. At times, the tips of mountain peaks are exposed above the ice surface, a formation known as a nunatak. Induced by gravity, these massive ice sheets or ice fields flow outward between the containing mountain peaks, forming outlet valley glaciers.

These valley glaciers are perhaps best described as "rivers of ice". The highest peaks of the underlying mountain range channel the larger ice flow into individual, linear "streams." Figure 1 shows a topological map of a small glacier section from the Juneau Ice Field in which the linear ice flows are clearly visible. These "streams" then flow downhill towards the ocean, thinning and picking up speed. Like conventional rivers, the ice sheet may separate into multiple branches, or different tributary glaciers may combine into a single, main flow.

The main area of a valley glacier is largely flat and covered with snow, defined at the edges by a chain of mountain peaks. The boundary between the moving ice and stationary mountain snow is often marked by vertical cracks in the ice, known as crevasses. The ice surface is also influenced by elevation changes in the rock bed, or changes in the glacier's path. Both of these result is significant undulations in the ice surface. In the extreme, an abrupt elevation change in the supporting terrain causes the glacier ice to tumble over the cliff in what is known as an "ice fall", before reforming at the lower level. At the end of the glacier's journey is an area called the terminus. Here the ice surface is exposed and warped by the terrain below. Many crevasses form, even in the center of flow, making traversal treacherous.

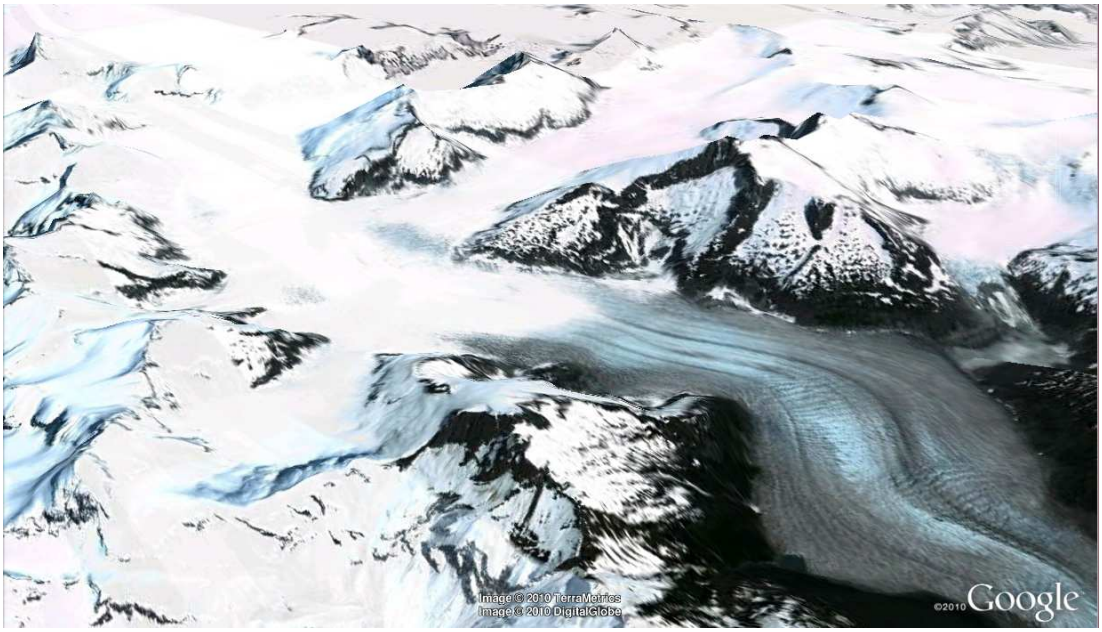
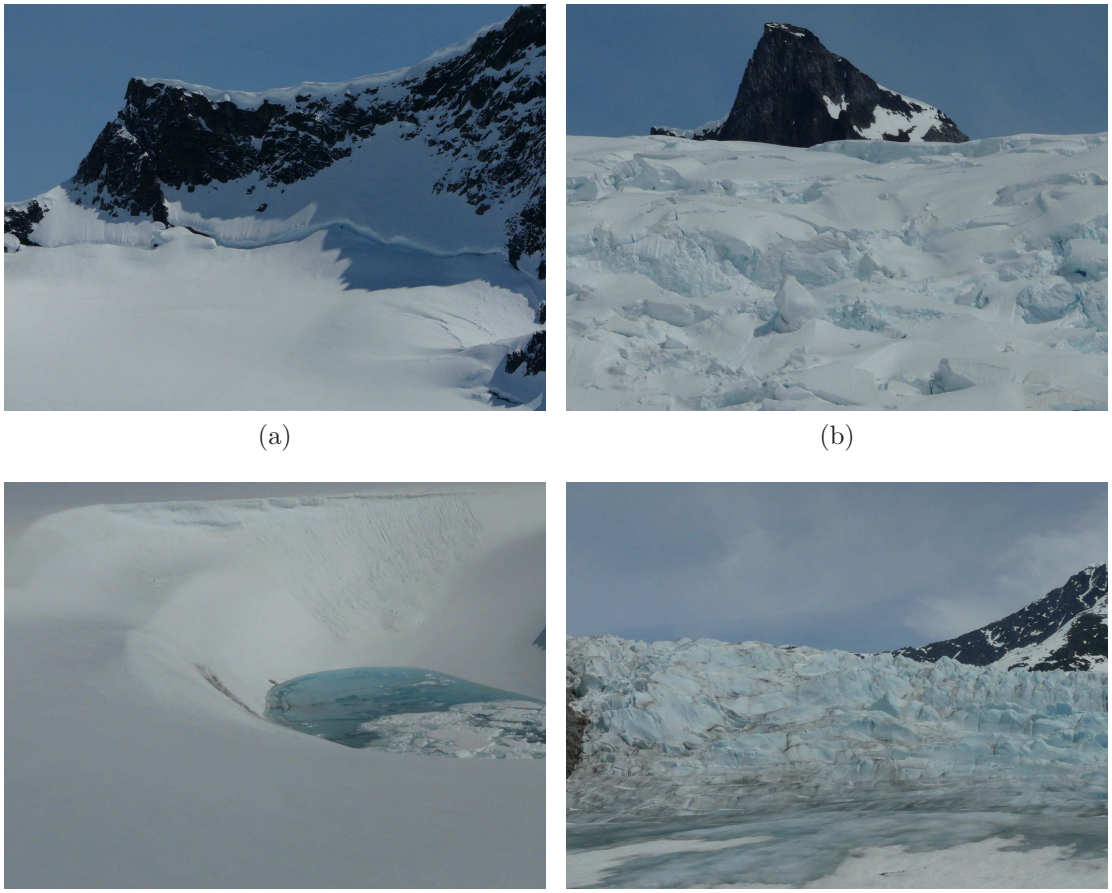


Figure 1: Topological map of a region of the Juneau Ice Field in Alaska. Individual "rivers of ice" are clearly seen, separated by mountain peaks.

Erosion mechanisms also shape the glacier surface. Winds create surface ripples with distinctive texture patterns, ranging in size from small ripples to dune-like sastrugi that reach over a meter high. Melt water also shapes the surface in the summer months, leaving behind erosion patterns in the snow, cutting meltwater streams through the surface, or collecting in a glacial lake basin. Figure 2 illustrates a number of these terrain features encountered on Mendenhall and Lemon Creek Glaciers in Juneau, Alaska.

As described, glacial environments present unique hazards that are visually distinct from those commonly found in desert or urban scenes. These hazards are predominantly "slope based," consisting of inclines and gentle perturbations in the glacier surface caused by changes in the rockbed, as well as more dramatic slope changes in the form of crevasses, lake basins, and sastrugi. Because a deployed robotic system will have a limited range of slopes that it is capable of traversing safely, any glacier-based hazard detection system must be capable of characterizing the terrain slope to ensure safe, autonomous operation.



(a) (b)

(c) (d)

Figure 2: Images from the Juneau Ice Fields in Alaska showing (a) a crevasse separating the flowing glacier from the stationary mountain ice, (b) an ice fall caused by a sudden elevation change in the underlying terrain, (c) a basin surrounding a forming glacial lake, and (d) the terminus of a glacier with exposed ice and many crevasses.

(a) (b)

Figure 3: Images from the Juneau Ice Fields in Alaska showing (a) an area of undulating terrain surface obscured by the low light levels of an overcast sky, and (b) an instance of near "white out" conditions in which the mountain range in the background is barely visible through the low clouds.

Weather patterns also negatively impact any deployed robotic platform. Aside from the obvious problems of extreme cold, high winds, and blowing snow, these regions often experience near complete cloud cover with low cloud deck elevations. This not only limits the available light, obscuring the few visible ground features in the snow-covered terrain, but it can also conceal mountain peaks behind the cloud wall. In extreme cases, a complete "white out" can occur in which the ground snow and clouds are indistinguishable from each other. Figure 3 illustrates instances of low surface contrast and "white out" conditions encountered in the Juneau Ice Field.

1.2 Glacial Robotics

Glacial regions present one of the harshest environments for mobile robots. Extremely low temperatures, high winds, and possible precipitation all impose significant design challenges. Despite this fact, there are several successful glacial robotic systems [128].

The CoolRobot effort out of Dartmouth College [68, 96] focused primarily on overcoming the environmental challenges with the construction of a glacier-worthy robotic platform. Thermal issues and power requirements were among the main

Figure 4: CoolRobot deployed at Summit Camp, Greenland. [68]

concerns, prompting the use of composite materials and solar panels. Additionally, proper wheel sizes and inflation pressures were identified for efficient traversal on the snow. The CoolRobot has been fielded successfully in glacial conditions, where it was able to operate autonomously for over eight hours. Figure 4 shows CoolRobot as deployed at Summit Camp in Greenland.

The University of Kansas developed the MARVIN I and MARVIN II rovers as part of the PRISM project [3,46], an effort to map subsurface features of the ice sheet using ground penetrating radar. The robotic chassis was constructed from a tracked ATV, designed by the manufacturer for high mobility in the snow and extended use in sub-zero temperatures. The chassis was then augmented with an automated control systems, computer hardware, and sensors contained within a ruggedized enclosure. The MARVIN rovers have been deployed for extended periods in both Greenland and Antarctica. Figure 5 illustrates the MARVIN I pulling a mockup radar array at the North Greenland Ice Core Project (GRIP) Camp.

The Nomad robot, developed previously for space application testing by Carnegie Melon, was "winterized" for deployment in Antarctica [5]. The intent of this mission was to autonomously locate meteor fragments on the Elephant Moraine, a flat sheet of blue ice littered with rocks ranging from pebbles to boulders. The rover was equipped

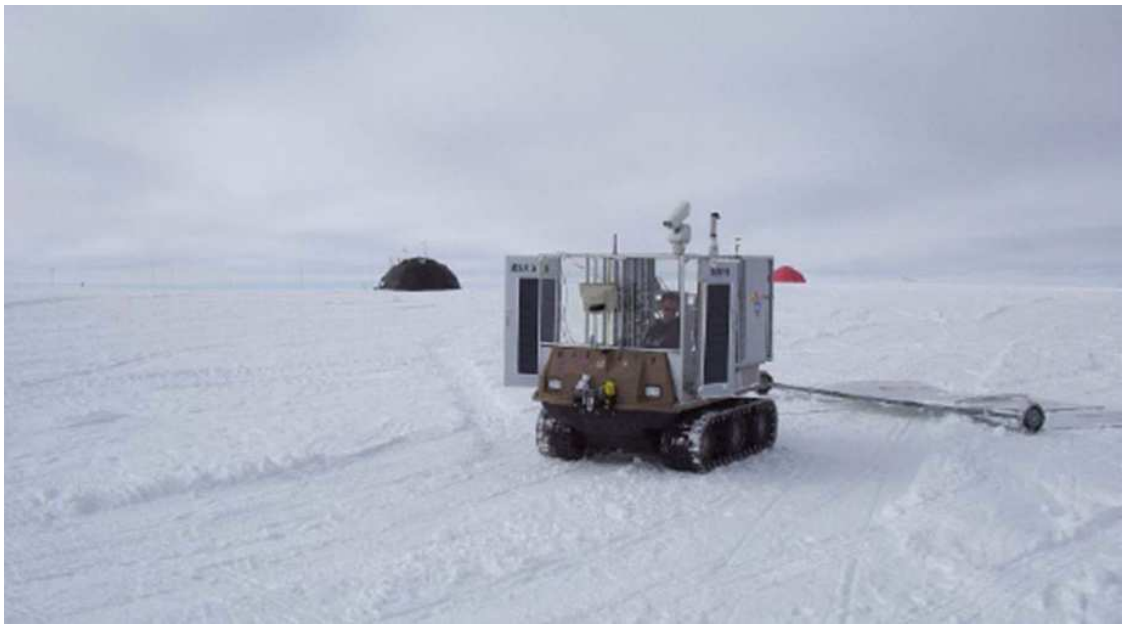


Figure 5: MARVIN I deployed at the North Greenland Ice Core Project Camp. [46]

with two different lidar systems, stereo cameras, a steerable monocular camera, as well as a wind-powered generator to fulfill the power requirements. The Nomad has been deployed to Antarctica several times, during which it has autonomously located and identified 5 meteorites. This rover has also been used in conjunction with sampling microbes in the Antarctica ice sheet [92]. Figure 6 shows the Nomad during a preliminary field test on Lake Mascoma, New Hampshire.

Additionally, a sensor comparison study was conducted as part of the Nomad field work, utilizing the wide array of sensing modalities present [117]. It was found that the lidar system produced many false readings during operation, especially during blowing snow conditions. Ultimately, the results of the sensor were filtered to provide confirmation of the absence of obstacles, rather than the more typical application of map building. The stereo vision system provided consistent results across multiple terrain types, but was negatively affected by different lighting conditions. Even under good operating conditions, the stereo pair had difficulty finding point correspondences due to the overall lack of contrast and features in snowy environments. The use of the monocular camera for detecting hazards, such as sastrugi, yielded some success, but

Figure 6: Nomad during an early field test on Lake Mascoma, New Hampshire. [92]

was again affected by lighting. An additional study was conducted using millimeter-wave radar to detect terrain obstacles [39]. While the radar system performance is not impacted by blowing snow, its sensing resolution is far coarser than camera images or laser scans.

These projects showcase the ability of the mechanics of a robot to survive the inhospitable climate of arctic environments. However, the autonomous navigation systems implemented on these robots are simplistic in comparison. Aside from a set of homeostasis sensors, the only other perception available on the CoolRobot is a GPS receiver. This limits the navigation system to simple GPS waypoint following. As a result, this robot can only be deployed in areas known to be safe and completely free of hazards. Similarly, GPS waypoint following is the primary navigational method of the MARVIN rovers. While this is a good fit for radar surveying tasks, it does limit the areas of possible deployment. The navigational system of the Nomad is more complex, consisting of a GPS waypoint system, an obstacle avoidance system based on the lidar data, a potential meteorite targeting system based on camera images, and a management system to gracefully switch between the different modes. However, due to the known flatness of the testing and deployment environment, little in the way of terrain assessment is required for successful navigation.

One final project of interest works towards automating a convoy of snowcats for the Association of European Research Establishments for Aeronautics (EREA) [11]. In contrast to the previously discussed projects, this effort focused on the navigational control system, and specifically vision-based terrain analysis techniques. The presented work used a vision system to locate the tracks of the previous vehicle in the low-contrast glacial environment, and was tested under many different lighting conditions. This project successfully developed algorithms tailored specifically for the difficulties inherent in visual glacial navigation.

1.3 Vision-based Robotic Navigation

As few examples of glacial-specific visual processing techniques exist [11, 117], an overview of various vision-based terrain assessment methodologies will be presented. This survey is meant to cover the major themes in this area of research, and highlight potentials for use in arctic navigation, rather than to provide an exhaustive summary of all vision-based research in the field of robotics. The scope of relevant research is further limited by considering those technologies designed to operate in unstructured, outdoor environments. As such, methods that rely on the detection of perpendicular lines [10, 12, 120], or assume a flat ground plane [7, 136], have been omitted. While these techniques perform well in indoor or urban environments, the basic assumptions of these methods are generally violated by cross-country or glacial landscapes.

1.3.1 Obstacle Detection

A common method for obstacle detection is to employ the use of a stereo imaging system. Once the stereo reconstruction is complete, segmenting the obstacles from the ground plane is not a trivial task. Several approaches assume the ground plane to be flat; anything that deviates from the planar surface must be an obstacle [7, 136]. However, in outdoor terrain, this assumption is easily violated. One solution defines an obstacle based on local size and slope properties [74]. Areas of shallow slope

are traversable, whereas tall regions of steep slope are likely obstacles. From this definition an efficient mechanism for locating obstacle patches is formulated, and connected obstacle patches are segmented into discrete obstacles.

In contrast, appearance-based methods attempt to segment images into hazardous and safe zones using visual properties rather than metric reconstructions. For example, rock segmentation routines have been developed for Mars applications [54]. The coloration of rocks versus the ground plane was identified as being statistically significant, enabling the use of a region-growing technique to separate the two in the image plane. Similarly, a method for vegetation classification has been proposed for cross-country autonomous vehicles [74]. Using sample data, several terrain classes are trained using a maximum likelihood strategy with a Mixture of Gaussians classifier. In this way, live vegetation, dry vegetation, and the ground could be separated based solely on color properties. This information was ultimately combined with lidar-detected obstacles to disambiguate tall grass from rocks and tree stumps. Similarly, a two-class unsupervised classification method distinguishes between image regions that are visually similar to the area in front of the rover and those that are different [114]. Under the assumption that the region directly in front of the rover is traversable, anything statistically different is marked as an obstacle. The use of alternative color spaces is mentioned explicitly, allowing the classifier to be more robust to shadowing effects. A similar methodology exists that employs self-supervised learning [61]. In terms of glacial terrain, such color-based methods could potentially be used to discriminate nunataks from the surrounding snow, or to differentiate between exposed ice at the terminus and soft snow. However, large regions exist that are entirely white and snow-covered, making color-based methods problematic.

Texture represents a second type of appearance attribute that can be utilized for terrain assessment. Texture, which is generally defined as the high frequency visual content of an image [138], often demonstrates material-specific properties. A roadway

segmentation system has been developed that detects anisotropic, oriented texture patterns [140]. The system consists of an anisotropic metric calculated in a small neighborhood around each pixel, and a clustering algorithm to segment the strongly anisotropic regions from the rest of the image. Although not a robotics application, an image segmentation system has been produced that is based not on color, but on texture [138]. Several examples are shown, such as a striped sweater, in which the texture is uniform within a single object, even though the coloration varies drastically. In terms of glacial scenery, the opposite can be true. Despite the snow and ice being uniformly white, differences in texture properties and orientations can be useful for classifying the terrain.

1.3.2 Localization

Localizing a robot means either determining the robot's pose within some *a priori* global map, or simply tracking the relative pose changes of the robot using the starting condition as a reference [9]. In either case, the robot's path through the environment must be tracked in a set of known units (e.g. meters and radians). This is known as a metric path reconstruction.

Using vision for localization, although desirable, poses difficulties due to the bearing-only nature of the sensor. Camera sensors are generally assumed to approximate the pinhole camera model, after lens distortions have been removed [51]. The pinhole model equations are listed in Equation (1) for reference. As can be seen, the pinhole camera model converts a three dimensional input position into a two dimension output pixel by normalizing all inputs by their depth. As the image plane is fixed relative to the camera focal point, each pixel may be viewed as a direction vector formed by connecting the camera focal point to the pixel location. As each pixel represents a direction, only the bearing of an observed world point may be extracted from an image. Metric information cannot be extracted directly from camera images.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} m_x \cdot X \\ m_y \cdot Y \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{1}$$

where $\left(\begin{smallmatrix} X \\ Y \\ Z \end{smallmatrix}\right)$ is the location of a point in 3D world coordinates, $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$ is the location of the same point in image pixel coordinates, f is the camera focal length, m_x and m_y are scale factors that convert pixel indices into metric distances on the image plane, and $\left(\begin{smallmatrix} u_0 \\ v_0 \end{smallmatrix}\right)$ is the principle point in pixel coordinates, ideally in the center of the image.

The general solution to localization using bearing-only observations is to incorporate metric information from an external source and use triangulation from multiple observations to perform the localization. A direct implementation of this approach uses distant mountain peaks as observations and a digital elevation map (DEM) as the source of metric data [16, 87, 137]. Given several peak sitings, from an omnidirectional camera or composite panorama, the triangulated location of the camera can be calculated. Similar techniques track the position of the sun [121] or star fields [103]. While these solutions are appealing, especially considering the presence of mountains in valley glacier regions, the cloud conditions of glacial environments often prevents the observation of mountain peaks. See Chapter 1.1 for a detailed discussion and example images.

The most prominent use of vision for localization revolves around multi-view geometry methods and the related simultaneous localization and mapping (SLAM) methods. To employ multi-view geometry using a single camera, multiple images are acquired at different points in time. Features in one image are then matched to features in the other images. With a sufficient number of point correspondences, the geometric transformation up to scale may be extracted by applying the epipolar geometric constraint [89]. Given an external source of odometry, such as wheel encoders or GPS, an estimate of the scale may be obtained as well.

In contrast, vision-based SLAM systems seek to estimate the 3D position of extracted image features. These position estimates are then projected into the camera image space of subsequent frames, and matched with image features in the new frame. The image space error between the projected landmark and the new image feature is used to update the position estimate of the landmark. Again, the camera system alone cannot produce a metric reconstruction, so external odometry sources are used.

Two major advantages arise from this approach. First, since the landmark positions are estimated in 3D world space, the 2D projections can be calculated directly, intrinsically satisfying the epipolar geometric constraints. Secondly, as the solution to the SLAM problem is a probability distribution, an estimate of the error of either the robot pose or the landmark positions may be obtained by marginalizing out the appropriate variable. These error estimates can then be projected into the camera frame, limiting the size of the search region for corresponding feature points. In contrast, propagating an error estimate through the 10th order root solving method required in the multi-view geometry approach simply is not practical [88].

Two major variants exist within the SLAM literature. The first uses a single extended Kalman filter (EKF) to perform the estimation, where the EKF state is a concatenation of the robot pose and the landmarks. However, by augmenting the EKF state vector with landmark positions, the complexity of the system grows as $O(n^2)$ in the number of landmarks, limiting the number of landmarks that can be maintained simultaneously [24]. A commonly used alternative is to employ a Rao-Blackwellized particle filter (PF) to estimate the robot pose [81], also known as FastSLAM. Each pose particle assumes the state is known exactly, decoupling the landmark distributions. In PF SLAM, each landmark consists of an independent low-dimension EKF, allowing the system to maintain a much larger database of active landmarks.

The SLAM-based methods have gained the most traction in the robotics community, and large-scale vision-based localization systems have been demonstrated [13,34]. For large-scale, outdoor environments, the PF SLAM approach had obvious advantages in terms of computation time and map database size. However, implicit in these algorithms is the ability to robustly find and match visual features between camera frames. Glacial terrain, with its low-contrast low-light snow-covered surface, does not offer the same kind of feature-rich scenery available in office buildings, or even in the Martian desert. Before such visual SLAM algorithms can be implemented, methods for extracting reliable visual features must be examined.

1.3.3 Terrain Mapping

Several aspects of terrain mapping must be considered: the source of terrain data, the data representation, and the potential uses of the map. Unfortunately, these topics are not necessarily independent; the type of data available will often affect the data representation decision, while subsequent uses of the map are far easier to perform with certain data representations. For mobile robotics, one of the main purposes of map building is to facilitate efficient path planning. On-board sensors observe the world around the robot, allowing paths to be planned around obstructions before they are encountered. Aside from planning, terrain reconstructions of glacial terrain can be useful for satellite sensor validation tasks. On-orbit sensing of quantities such as surface solar radiation are dependent on measuring quantities reflected from the Earth's surface. In order to interpret the raw sensor readings, an estimate of the terrain surface is required [124]. More accurate terrain maps will lead to better output data products. Another possible use includes improved estimates of the glacial ice mass balance [48]. The periodic construction of high quality glacial maps could allow scientists to better estimate the change in ice volume between seasons or years.

Perhaps the most common mapping representation in mobile robotics is the occupancy grid. With an occupancy grid, the environment is segmented into fixed-sized squares and the probability of an obstacle existing within each square is encoded into the map [35,63]. For three dimensional mapping, an analogous system based on dividing the environment in cubes has been used for unmanned aerial vehicles (UAVs) [55]. Ranging sensors, such as sonar and lidar, are ideal for this type of mapping, although vision-based systems have also been used to produce similar results [104]. These types of maps have also been used in outdoor environments when obstacles tend to be discrete in nature (e.g. rocks, trees, and tall grass) [1]. Path planning tasks are easily performed in occupancy grids, with such classic algorithm examples as A* [50]. However, as previously described in Chapter 1.1, obstacles in glacial environments are slope-based rather than discrete. While some threshold slope value could be applied to convert high slope areas into discrete obstacles, terrain slopes are a three dimensional conception; formations such as ridge lines and saddle points present uniquely different slopes based on the orientation of the observer. The situation-dependent nature slope-based traversability assessment makes the use of occupancy grids in glacial environments less than ideal.

An irregular triangular mesh (ITM) is an alternative mapping representation in which the terrain surface is approximated with a sparse set of tessellated triangular patches. This allows any path planning strategies to estimate the rover-experienced terrain slope of a particular path, although these estimates are piece-wise planar and slope discontinuities exist at the triangle boundaries. These types of maps are generally constructed from stereo vision data [62, 71] or laser scan data [97], which provide dense 3D estimates of the local terrain surface. In these systems, either all data is used within the ITM, in which case the terrain representation grows linearly with the number of samples [118], or data decimation strategies must be applied to extract sparse triangular elements [66, 97]. As previously discussed in Section

1.3.2, SLAM-based localization systems generate a sparse set of terrain surface points, complete with uncertainty estimates. However, the opposite problem occurs when using ITM with sparse data. Terrain areas with incomplete or insufficiently sampled points will be modeled as a single plane, despite the presence of complex terrain features. Further, the ITM does not inherently consider the sample uncertainty, reducing the potential usefulness of this representation.

In the field of geostatistics, a modeling method known as "Kriging" is common place [17]. Kriging estimates the terrain elevation at arbitrary locations using weighted sample information in the vicinity of the query point. The weights are derived from a covariance function, which returns the estimated correlation between two geographic points. Sample data is generally used to derive a variogram, which relates the amount of expected terrain elevation variation with respect to the distance between samples. One of a number of functional models is then fit to the variogram and used as the covariance function [112]. This system is a specialization of a statistical random process, and, as such, is capable of incorporating measurement uncertainty in a rigorous way. Additionally, the query points may be requested at an any resolution, allowing digital elevation maps (DEM) or ITMs to be constructed at arbitrary resolutions or allowing the creation of multi-resolution maps from a single representation. In geostatistics, the sample locations are generally selected manually to improve the reconstruction quality [123], although there is no mathematical requirement to do so. In robotics, the related field of Gaussian processes (GP) have been utilized to perform a terrain reconstruction using multiple 3D lidar scans of a single scene [118, 119].

Finally, other vision-based reconstruction methods are available that do not require the use of 3D surface samples. Make3D [101] creates 3D reconstructions of arbitrary scenes from a single image. The image is segmented into small, homogeneous "superpixels" that are then fitted to a plane. The system employs a Markov Random Field (MRF) to estimate the plane parameters (depth and orientation) for

each superpixel. The MRF is trained using a set of images with corresponding laser scan data to act as ground truth. The system also incorporates human-inspired monocular cues, such as convergent lines, texture variations, and defocus to break the ambiguity of different possible 3D structures, and return a single, feasible scene model. Still other methods attempt to recover scene geometry from texture variations [41, 56, 126], or shading effects [141]. The incorporation of surface texture into the reconstruction method is of particular interest. As noted previously in Section 1.3.1, surface texture can provide visual cues as to the nature of the terrain surface.

CHAPTER II

ROBOTIC INFRASTRUCTURE

In order to test any produced glacial navigation algorithms, a certain amount of robotic infrastructure must first be developed. Standard wheeled robotic platforms, such as the iCreate or the Pioneer, are not equipped to travel on ice and snow, and any vision-based methods will require sets of sequential images from analogous glacial environments as input. To that end, a set of snow-worthy robotic platforms have been designed and constructed. These platforms have been fielded several times on glacial terrain, during which data was logged for extended traverses. Finally, a visually faithful simulation environment was developed to ease the testing requirements of developed software, as well as to allow ground truth data collection. The following sections describe the rover design, detail the terrain and data collected at each field test, and discuss the creation and evaluation of the simulation system.

2.1 Robotic Platform

The previous arctic robotics projects, described in detail in Chapter 1.2 showcase the ability of the mechanics of a robot to survive the inhospitable climate of glacial environments. However, each of these projects involves the construction of a single, expensive robotic agent. Such an approach is not practical for the development of multi-agent systems, where potentially dozens of robotic agents will be utilized. As many rovers will be required to create a usable sensor network, each unit must be inexpensive. This pushes the design away from centimeter accuracy GPS and military-grade IMU sensors, and towards consumer-grade sensing technologies. Further, high terrain mobility must be emphasized in the design and construction. While much of the rover's time will be spent in the flat, central regions of the glacier, the project goal

is to construct a system capable of traversing the widest range of expected terrain possible. Typically the areas of most interest to scientists occur at the extremes of the environment. Collecting data about a forming glacial lake requires descending into the surrounding basin, while investigating the glacier-mountain boundary requires ascending steep slopes while identifying and avoiding crevasses. Due to the extreme nature of the terrain, autonomous exploration of large expanses of the terminus is probably not possible, but deployment in the upper regions should be feasible. To this end, three prototype mobile weather sensor nodes were constructed as part of this research, with an emphasis on low-cost sensing and all-terrain mobility [132]. Details of the design iterations of the SnoMote rover follow.

2.1.1 SnoMote Mk1

A 1/10 scale snowmobile chassis was selected for the first prototype platform, endowing the rover with an inherent all-terrain drive system. The platform was modified to include an ARM-based processor running a specialized version of Linux. The motherboard offered several serial standards for communication, in addition to wifi and bluetooth. A daughterboard provided an ADC unit to interface with the on-board science package, and PWM outputs for controlling servos. The inclusion of wireless communication protocols allows flexibility in the placement and packaging of the various sensors in the system, reducing the need for weather-proof connections. All control electronics were housed in a water-resistant compartment in the rear of the platform, while the camera system and GPS receiver were placed in sensor-appropriate locations. The drive system was modified to accept PWM motor speed commands, leaving the manufacture-supplied motor and gearbox intact. Steering control was provided by a weather-resistant high-torque servo motor. For ground truth position logging, a commodity GPS unit communicates to the processor via the bluetooth interface, while robot state and camera images are sent directly to an external control

Table 1: Major system components of the SnoMote Mk1 prototype rover

Component	Manufacturer	Relevant Specifications
Embedded Processor	Gumstix	400MHz ARM processor, 802.11g, bluetooth
Micro-controller	Atmel	SPI serial, 10-bit ADC, PWM, General I/O
Drive Motor	New Bright	bushed DC, 50W max
Motor Controller	Solarbotics	H-Bridge, PWM output, 4A output max
Steering Servo	Hitec	45 oz-in torque
Camera	Axis	640x480 image resolution, color, wifi, MJPEG stream
GPS	Globalsat	10m rated accuracy, bluetooth
Temperature	Analog Devices	$\pm 1.0°C$, 3.3V output
Relative Humidity	Honeywell	$\pm 2\% RH$, 3.9V output
Pressure	Freescale	$\pm 1.5 kPa$, 4.8V output
Accelerometer	Analog Devices	$\pm 0.01 g$, 2.6V output, 3 axis

computer via the wifi link. To simulate the science objectives of the mobile sensor, a weather-oriented sensor suite was included that measured temperature, barometric pressure, and relative humidity. Table 1 lists the major system components for the Mk1 rover.

The Mk1 platform was deployed on Mendenhall Glacier, Alaska in June/2008, as shown in Figure 7. The three Mk1 rovers were driven manually to assess the mobility performance in the different snow conditions present. During these traverses it was discovered that the platform suffered from stability issues. Due to the narrow track footprint in the rear, the chassis would often roll sideways when attempting to navigate perturbations in the snow surface. Additionally, the snowmobile would sink in the fresh snow, causing the DC drive motor to stall from excess torque. Due to

(a) (b)

Figure 7: (a) One of the Mk1 rovers deployed at Mendenhall Glacier in Juneau, Alaska and (b) all three rovers just before being loaded into the charter helicopter after testing.

the chassis limitations, only short traverses were performed in selected locations. Due to the design inadequacies of the initial, manufacturer-supplied chassis, it was clear major mechanical revisions would be necessary before redeployment.

2.1.2 SnoMote Mk2

The stability issues of the Mk1 rover stem from the basic design. Standard snowmobiles operate with a motor driving a single track system located in the central rear body of the chassis, which is guided by two runners located near the nose of the chassis on either side. Typically this design lends itself for maneuverability when combined with a rider who shifts their weight to restore balance. Without the physical presence of a rider, the standard design lacks the ability to systematically redistribute weight and is vulnerable to toppling.

A dual tread drive train system was implemented in response to these problems. The new system improved performance over the original design in two major ways. First, it nearly doubled the surface area in contact with the snow. By reducing the applied surface pressure, issues of sinking and traction loss in soft snow are reduced. Secondly, by modifying the rear sector of the chassis, the surface contact footprint

was converted from a nearly triangular pattern to a more rectangular shape. This greatly improves the stability characteristics of the platform, reducing the likelihood of roll-over.

A rear wheel drive and front wheel steering system was selected over other ideas such as a differential system because of its level of simplicity. Installing an additional track and harness alongside the existing one would require some external retrofits, but ultimately required only an extension to the drive shaft, leaving the existing motor and gear system intact. In contrast, a differential drive system would require either a mechanical differential and clutch, or a second motor, gearbox, and drive shaft operating in parallel.

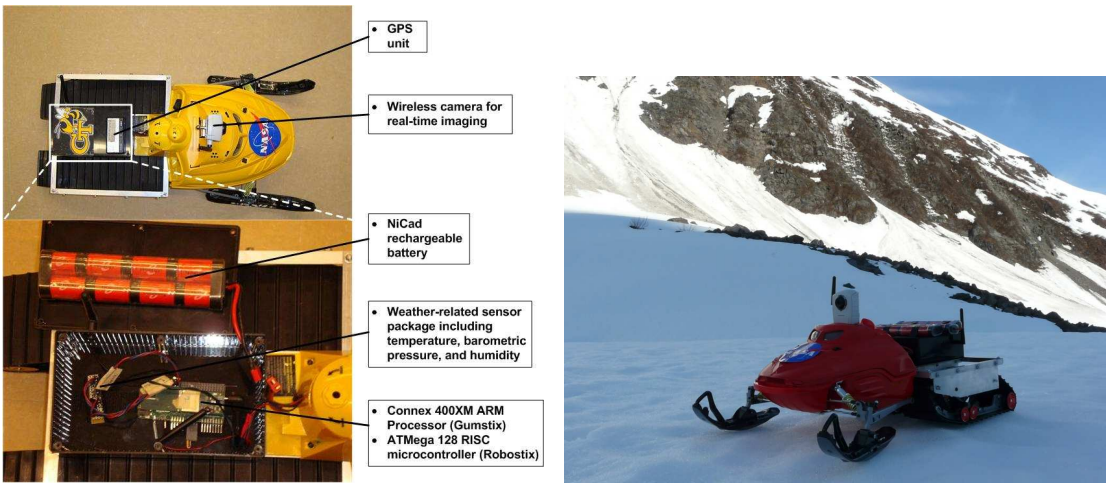
The dual tread system required a complete redesign of the rear section to accommodate the two tracks. A 3D model of the proposed upgrades was created using a computer aided design (CAD) software package. The software provided the ability to examine individual components and test the assembly process by generating a visual preview of how each part mates with the existing platform. From these virtual representations, necessary adjustments could be made without the cost or material consumption associated with physical design errors. The dimensions of the treads were taken and implemented into the CAD software to design a suitable harness. The harness also needed to be integrated with the remaining snow mobile platform, requiring that some reinforcements be applied at the attachment point. These recommendations were all integrated in the three dimensional model and adjusted accordingly. The model provided the capability to determine a new location of the motor and drive shaft that would work properly with the new design. Once a final design was agreed upon, fabrication of the supplemental parts was commenced. The successful completion of the new tread harness design securely held the tracks in place and the drive shaft was smoothly guided through both track systems and driving motors. Figure 8a shows the final dual track design implemented on the prototype rover.

(a) (b)

Figure 8: (a) The dual tread drive train system improves the rover's stability and traction in soft snow. (b) The aluminum double-wishbone suspension design increases the ski separation by 30%, introduces tunable spring-over-oil dampers, and provides increased travel distance while maintaining the proper ski orientation.

The original platform also suffered from steering deficiencies. The stock steering linkages of the snowmobile lacked the necessary rigidity to effectively maneuver through the depths of snow present as the test site. At the same time, the stock suspension system was too stiff, directly translating surface changes into body roll, instead of compensating ride height. To correct these issues, the stock single-arm plastic linkage was replaced with an aluminum double-wishbone design. The new parallel linkage system extended the ski separation by 30%, and provided increased travel distance while maintaining the proper ski orientation. The corresponding tie rod was similarly lengthened, connecting the steering servo to the steering arm at the ski. The newly installed suspension additionally included a spring-over-oil damper system that was easily adjustable in the field. This allowed the stiffness to be tuned to the quality of the terrain. Figure 8b shows the new suspension system.

In January/2009, a test site near Wapakoneta, Ohio was selected to validate the performance of the new chassis in anticipation of future glacial testing, shown in Figure 9. The improved chassis performed well during the tests, never rolling, even when negotiating a path between rocks up a 20° slope. While it was still possible for the chassis to loose traction, especially in very soft snow or up steep inclines, the



(a) (b)

Figure 9: (a) Component diagram of the completed SnoMote Mk2, and (b) the platform as deployed at Byron Glacier, Anchorage, Alaska.

new drive motor was never forced into a stall condition. However, one unexpected observation from these tests was that the control computer, which consisted of a consumer-grade laptop, ceased to operate when its temperature dropped below 20°F. The SnoMote electronics, however, were unaffected by the cold.

2.2 Field Trials

During the course of the preliminary research, several field trials and data gathering missions were conducted. The following sections outline the purpose of each trip, the terrain encountered, and the data collected.

2.2.1 Arikaree Glacier, Colorado - March/2007

Arikaree Glacier is a small cirque glacier formed in a mountain basin in the Niwot Ridge, near Boulder, Colorado. The mass balance of Arikaree has been studied since 1963, and is currently under observation as part of the Long-Term Ecological Research (LTER) program, sponsored by NSF [72]. The main research station, located on the Niwot Saddle, is accessible via an unimproved roadway. Arikaree is located approximately 5 km from the research station, accessible only by snowmobile.

(a) (b)

Figure 10: Representative terrain images from the March/2007 field visit to Arikaree Glacier, Colorado.

The purpose of the March/2007 trip was to assess Arikaree as a potential test site for further research, and to collect firsthand imagery of an analogous arctic environment. As a cirque glacier, the site was surrounded by relatively close mountain peaks, visible in almost every direction. The terrain consisted of hard-packed snow that clung to the mountain face nearly to the peaks. As such, significant terrain inclines were present at the glacier's edge. The central basin consisted of less than 1 km^2 of flat terrain. Figure 10 illustrates typical terrain imagery from Arikaree.

Due to the travel logistics, equipment was kept to a minimum. Images of the terrain were filmed from various angles, and under changing lighting conditions. Additionally, close-ups of the terrain texture were captured. In total, the Arikaree Data Set consists of over 20,000 video frames.

2.2.2 Mendenhall Glacier, Alaska - June/2008

Due to the imposing travel logistics of transporting equipment to Arikaree, a new test site was selected for 2008. Mendenhall Glacier is part of the Juneau Ice Field, the fifth largest glacier system in North America. As part of the Tongass National Forest, the Mendenhall Glacier is visited by almost half a million people annually. In addition, the Mendenhall Glacier is the subject of ongoing scientific research by

the SEAMonster Project [15]. The surface of Mendenhall is inaccessible by roadway. However, Juneau is home to an extensive industry of charter helicopters servicing mining operations and tourists. These helicopters are capable of transporting six passengers and a substantial equipment payload, reducing the total travel time to under an hour.

However, helicopter travel to glacial areas is heavily dependent on the weather conditions, particularly low cloud deck heights. This presents a dangerous "white out" situation for the helicopter pilot in which the snow-covered peaks, the ground of the landing site, and the sky are all indistinguishably white. During the June/2008 tests, this condition limited the travel options to the lower section of Mendenhall, near the terminus.

The site surface is visually flat and covered with snow, though there are sections of the terrain where the underlying ice sheet is exposed. Despite the flat appearance, the snow varied in depth from a few centimeters to over a meter. This snow was deposited recently and was quite soft. Upon arrival at the site, a test area was explored with ice-axes to ensure it was safe. Cracks in the underlying ice, called crevasses, are often completely concealed by surface snow. Figure 11 shows the types of terrain encountered at Mendenhall Glacier.

Using the three Mk1 rovers, a set of short traverses were performed in selected locations. During these traverses, the local temperature, barometric pressure, relative humidity, GPS location, and camera images were all logged at 2 Hz and timestamped to ensure proper off-line reconstruction and analysis.

2.2.3 Indian Creek Lake, Ohio - January/2009

A test site near Wapakoneta, Ohio was selected to verify the performance of the new chassis in anticipation of future glacial testing. The site was blanketed with 8-12 inches of fresh snow next to the frozen Indian Creek Lake. Several long traverses were

(a) (b)

Figure 11: Representative terrain images from the June/2008 field trials on Mendenhall Glacier, Alaska.

conducted that transitioned from land to lake several times. During these traverses, the GPS location and camera image were logged at 15 Hz and timestamped. The lake bank consisted of irregularly spaced large rocks, between which large amounts of snow had collected, forming a drivable incline between 10° and 30°.

2.2.4 Mendenhall and Lemon Creek Glaciers, Alaska - May/2009

Mendenhall Glacier was the subject of the May/2009 field tests as well. Favorable weather conditions allowed multiple tests to be conducted. Several test sites were selected across Mendenhall in order to test the system in a variety of glacial terrains. Site A was located at the top of Mendenhall. Several mountain peaks were visible above the flat glacier surface, the closest being over a kilometer away. Site C was on the upper plateau of the terminus. Here the underlying ice is exposed and the terrain is characterized by small, rolling hills one to two meters in height. Some crevasses are present in this area, and melt water pools in some of the small valleys. Site D was at the top of the northern branch. This area is completely covered with over a meter of soft snow and is largely flat for several kilometers in any direction. Site E was located at the lower edge of the northern branch, near a bend in the glacier. Again, the site is completely snow covered, but is much closer to the mountains. Due to

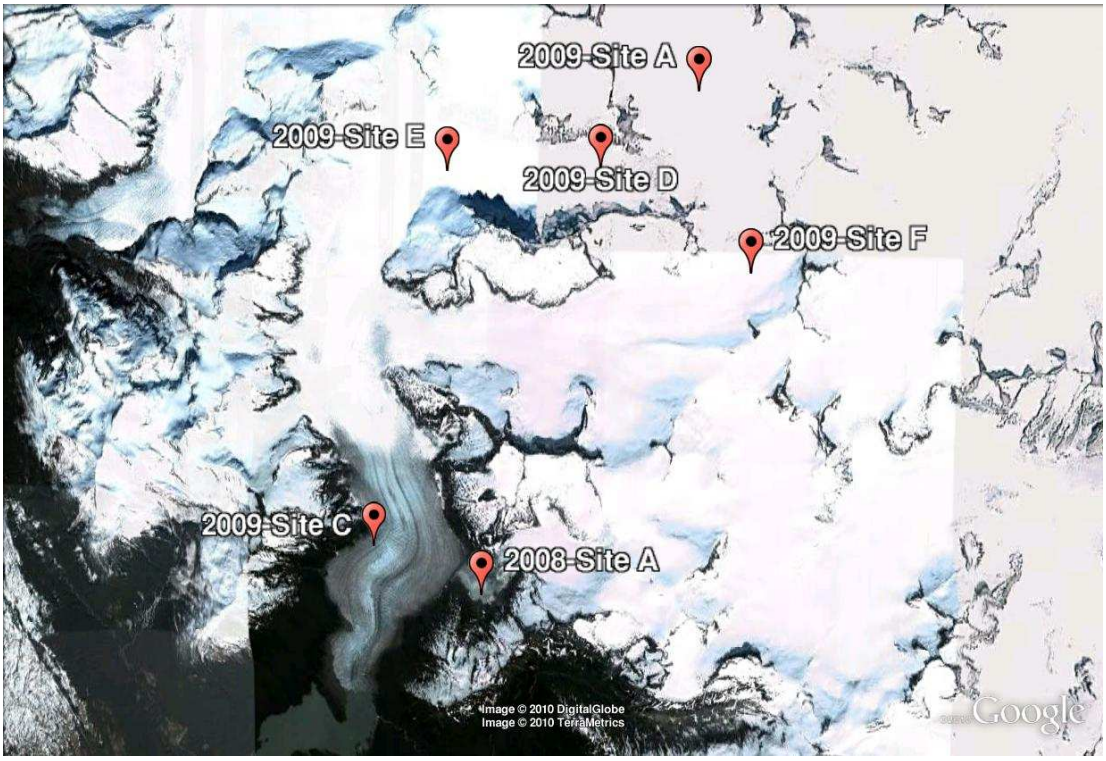
the proximity of the Mendenhall Tower peaks and the bend in the path, the terrain exhibited large-scale undulations. Site F was at the top of the southern branch, on the opposite side of the Mendenhall Tower Peaks from Site D. This area resembled Site D, and was covered with soft snow with visible distant peaks.

The weather also permitted travel to a second glacier at a higher elevation. Site B was located on Lemon Creek Glacier, a site monitored by Juneau Icefield Research Program (JIRP) since 1946 [57]. The area was marked by the presence of a forming glacial lake, and several mountain peaks an accessible distance away. Figure 12 shows the location of each test site on the glacier, as well as images of the typical terrain.

At each test site, a set of salient still images were acquired, and video sequences were recorded from the rover's onboard camera. Individual maneuvers included small closed loops on flat terrain, long linear runs up significant terrain inclines, and switchbacks running down the side of mountain peaks. In total these recordings represent over 50,000 individual frames, and account for over a kilometer of traversal distance. In addition to the collected images, GPS data, weather measurements, and accelerometer data were all recorded and timestamped for future analysis.

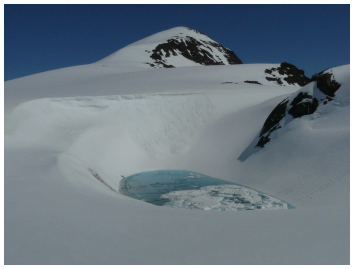
2.2.5 Byron Glacier, Alaska - May/2010

A narrow testing opportunity in May/2010 allowed a one-day field trial on Byron Glacier near Anchorage, Alaska. The terminus area of Byron Glacier is accessible by foot, and exists within a narrow valley surrounded by mountain ridges. The snow conditions differed significantly from the other testing sites, consisting of a layer of refrozen snow. This outer layer tended to be firm and slick, more similar to ice than the soft snow of Mendenhall and Lemon Creek. Consequently, the ribbed track system and steering skis of SnoMote rover did not offer the same traction advantages when used on a deformable surface. Despite this, the rover was still maneuverable and able to traverse an upward slope approaching 20°.



(a) Site Map

(b) Site A



(c) Site B

(d) Site C

(e) Site D

(f) Site E

(g) Site F

Figure 12: (a) A map of the relative position of the various test sites on Mendenhall Glacier during the May/2009 field tests (Lemon Creek Glacier Test Site B not shown). (b)-(g) Sample images from each test site illustrate the typical terrain.

(a) Site A (b) Site B

Figure 13: Representative terrain images from the May/2010 field trials on Byron Glacier, Alaska.

Two different test sites were utilized during the day's testing. Site A was located near the center of the valley, near a saddle point in the terrain topology. Significant terrain slopes existed within a short traverse of the base location. Site B was was much flatter, located beside a small melt-water stream. The remains of a recent avalanche marked the southern testing limit. Figure 13 shows images of the typical terrain from each test site.

2.3 Simulation System

Unlike more traditional environments, preforming field tests in glacial conditions is far more involved. Schedules must be arranged, equipment shipped, and helicopters chartered. Field work is thus more expensive and time consuming, resulting in fewer opportunities to test algorithms. Even during the conducted field tests, measuring the ground truth for the environment is a challenge. The rover traversed several hundred meters at each test site, making area laser scans of the terrain impractical for the limited duration testing.

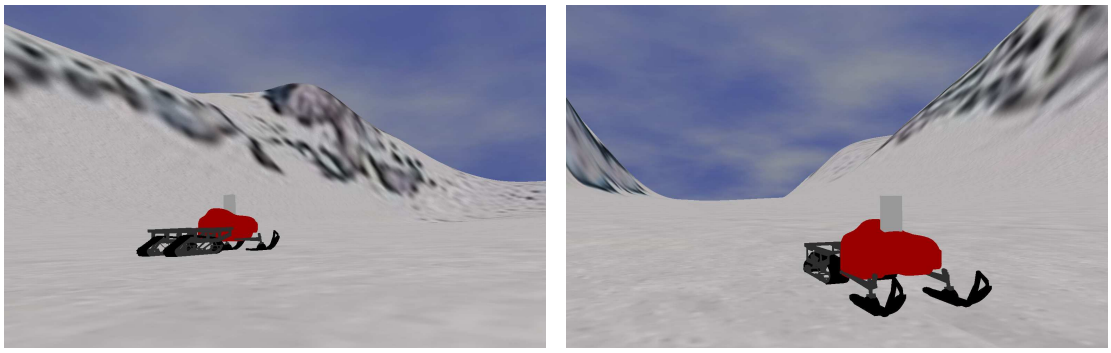
Simulation is often considered as a means of addressing facets of this challenge. However, due to the generally poor rendering quality of typical robotic simulation

environments, simulation is rarely, if ever, used to evaluate vision-based algorithms. Despite this challenge, a simulation system has been developed specifically for testing visual navigation algorithms. This simulation system, which uses Gazebo [44] as its base, has been extended to provide a visually faithful environment including realistic large scale terrain, local scale hazards, and background imagery. The following describes in detail the methods used and extensions created in the development of the visually faithful simulation. Additionally, an approach for evaluating the efficacy of the constructed simulation system is also presented, which makes use of algorithm-specific performance metrics to compare the simulation to the real environment [134].

2.3.1 Large-Scale Terrain

One of the first aspects that must be handled in outdoor simulations is the creation of the large-scale terrain. Within Gazebo, a heightmap image is used that encodes relative elevations as pixel intensities. 3D modeling programs, such as Blender [42], generally supply manual methods for working with heightmaps. This involves digitally sculpting the heightmap from a flat mesh using various tools. This allows an enormous amount of discretion in the behavior of the terrain, but requires time and skill to produce realistic heightmaps. Once completed, the heightmap must still be "painted" to produce the overlay texture, again requiring time and skill to produce good quality results. Procedural creation, on the other hand, automatically generates a random terrain based on a few user parameters. This allows relatively unskilled designers to produce good quality results, while simultaneously generating a texture map.

Perhaps the best method is to use elevation data from the desired test site. Recently released digital elevation models (DEM) from the Shuttle Radar Topology Mission (SRTM) [36] are available for most of the globe at 90m resolution, while the Lunar Orbiter Laser Altimeter (LOLA) instrument on-board the Lunar Reconnaissance Orbiter (LRO) is in the process of gathering similar data for the Moon. These



(a) (b)

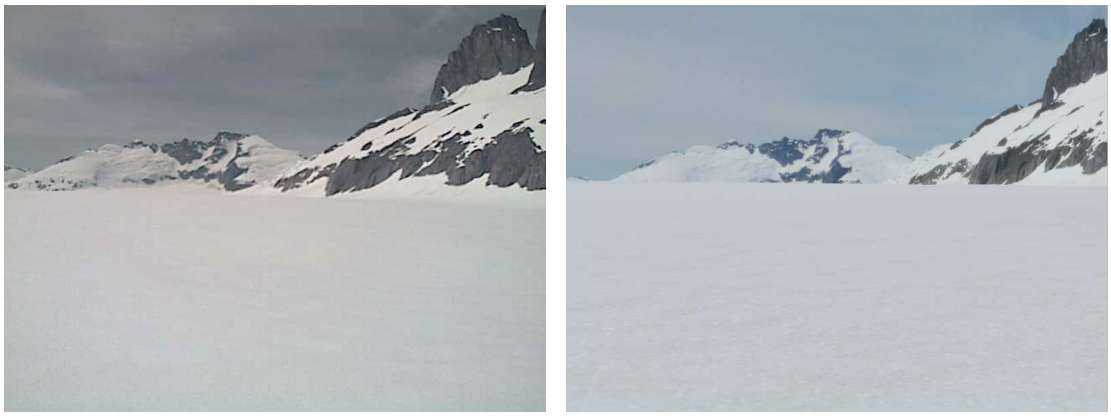
Figure 14: Sample images from a simulation of the upper north branch of Mendenhall Glacier generated using DEM models from the SRTM data set and satellite imagery.

data products can easily be converted into heightmaps usable by Gazebo. Similarly, geo-registered satellite imagery can be obtained from sources such as LandSat for terrestrial locations, Lunar Reconnaissance Orbiter Camera (LROC) on the LRO, or HiRISE for Martian terrain. These images provide the large-scale coloration needed to render the terrain accurately. Figure 14 illustrates a sample simulation system of Mendenhall Glacier in Alaska using these techniques.

2.3.2 Backgrounds

The large-scale terrain must be textured using overhead-view images, such as those obtained from satellite or aerial photography. This is appropriate for terrain on which the robot will be traversing, using the projective camera model to render the ground from any angle. However, much of what is visible by an *in situ* rover may be classified as background images; mountains, sky, and other distant features differ significantly in visual appearance at ground level versus an overhead view. Such details are important to simulate for vision-based algorithms, as these aspects are often hard to separate from the ground plane of interest. Alternatively, in the context of lunar robotics, such things as star trackers could be implemented to aid navigation if they could be rendered accurately within the simulation.

One effective way of generating realistic background rendering is to employ the



(a) (b)

Figure 15: (a) A image acquired from the on-board rover camera during a field trial on Mendenhall Glacier, and (b) a corresponding imaged generated by the simulated world within Gazebo.

use of a "skybox." A skybox is simple a large box containing all of the simulation elements. A texture is applied to the inside of this box, allowing background elements, such as mountains in the distance or clouds in the sky, to be generated as a set of static images. It is possible to render the skybox as if it were following the camera, so the background elements are always equidistant from the camera. This is a good approximation if the background is far from the camera relative to the distance traveled by the rover, such as stars or mountains, which are tens of kilometers away.

For the case under consideration, a skybox is used to render in photo-quality the mountains surrounding the glacier test site, using the DEM to generate the driveable terrain only. While at the test site, a number of high resolution images were taken spanning the entire horizon. Using photo-stitching software, Hugin [20] in this case, these images were transformed into a single panoramic image. This image was then divided into four segments, one for each of the vertical skybox sides. An image of the sky was used for the top plane of the skybox, while the bottom plane is concealed by the DEM terrain. Figure 15 shows a visual comparison between the final simulation rendered from the rover's perspective with that of a real image acquired from the onboard rover camera.

2.3.3 Evaluation Method

When comparing the visual rendering quality of a simulation system versus images of the real terrain, it is difficult to perform anything other than a qualitative comparison. Further, determining how real is "real enough" is not obvious, and is generally dependent on the combination of task and visual algorithms employed. Ideally, the output of the visual algorithms in question should provide similar results when applied to the simulation system compared to those obtained from the real environment. Therefore, the simulation system may be deemed "sufficient" for a specific visual algorithm and corresponding performance metric if the difference in the results from the two environments are not statistically significant. Further, the quality of the simulation system can be derived from the number and variety of visual algorithms that perform properly within the simulation. This methodology is used to evaluate the simulation quality for each visual algorithm under consideration, before results obtained from the simulation system are considered.

CHAPTER III

VISUAL TRAVERSABILITY ASSESSMENT

As described in Chapter 1, one requirements for an autonomous mobile arctic sensor network is a hazard-avoidance strategy capable of operating in real-time on commercially-available embedded hardware. While little research has been devoted specifically to obstacle detection and avoidance in glacial environments, a large amount of work does exist for desert and urban environments.

A majority of this work relies solely, or in part, on the use of laser range scanners to detect and characterize obstacles. The H1ghlander and Sandstorm autonomous vehicles from CMU fuse multiple lidar and radar systems to estimate the traversability cost of near-by terrain [116]. Stanley, the winning entry to the DARPA Grand Challenge in 2005, uses laser range data to determine the traversable area near the current vehicle position [19, 113]. This area is then used to train a color model, allowing the entire image to be classified as drivable or non-drivable. In a similar manner, the MuCAR uses laser scan data to generate a set of possible vehicle trajectories, then incorporates visual cues to eliminate non-traversable paths [75]. Other work focuses on finding visual road features, such as curbs and lane markers, which are also visible in lidar data [67, 107].

However, the use of laser range scanners within the context of a multi-robot deployment in glacial terrain is far from ideal. Common 2D line scanners, such as the SICK LMS 291 [28], are expensive (in excess of $5k), offer a limited sensing range (80m), and consume significant power (20W). Further, while a single line scan is capable of detecting discrete obstacles, it does little to characterize the terrain slope, which, as described in Chapter 1.1, constitutes the major hazard in glacial

terrain. 3D scanners, constructed of either a 2D scanner on a rotating base or as self-contained units, can better map the terrain, but suffer from increased cost and power requirements, slower scanning times, or severely limited range (30m for the Hokuyo UTM-30LX [27]). Finally, early tests of laser range scanners in Antarctica indicated that the reflectivity properties of snow and ice translated into significantly increased noise in the laser scan output [117].

Based on the difficulties of active sensing techniques such as lidar or radar, the exploration of terrain assessment methods have focused on the use of vision. Specifically, a terrain traversability map has been created, inspired by techniques used for desert or Martian terrain [54]. This traversability map incorporates color and texture cues from the image to generate the estimated terrain traversability of each pixel location. Using the traversability map as input, it is shown that a simple, reactive control strategy is sufficient to avoid the terrain hazards present in glacial environments.

3.1 Region of Interest

As discussed in Chapter 1.1, the weather conditions in glacial regions can often obscure the ground-sky boundary, a condition known as "white-out." Before any meaningful analysis of terrain obstacles can be performed, the foreground region must first be segmented from the acquired image. This serves to eliminate image features, such as background mountains or cloud features that could interfere with obstacle assessment routines, as well as focus subsequent processing on a smaller, targeted region of interest. Potential foreground segmentation methods are presented below, as well as a novel horizon line detection process that has been tailored to work on glacial images [130]. All methods are tested against a set of hand labeled images consisting of samples taken from each of the ten different field trial sites.

3.1.1 Adaptive Histogram Threshold

One project, sponsored by the Association of European Research Establishments in Aeronautics (EREA), uses vision to extract the foreground from glacial images as part of an automated snowcat convoy system. In order to remove the unwanted regions from arctic imagery, the EREA project [11] proposed using an adaptive histogram threshold to separate the foreground from the rest of the image. It is assumed that the majority of the image is filled with the snowy region. Consequently, in the histogram of the image, the largest peak should be associated with the grayscale values of this region. An adaptive threshold based on the boundaries of this peak is then used to separate the region of interest from unwanted objects and areas. Computationally, this method is very efficient, requiring only a single pass through all of the pixels to construct the histogram, a linear search within the histogram space, and a second pass through all of the pixels to apply the threshold values. This technique represents the only known prior art in the area of glacial foreground segmentation.

The adaptive histogram threshold strategy has been applied to the two sample images shown in Figure 16. The first image is from the June/2008 data set on Mendenhall Glacier. In this image there is a distinctive foreground-background horizon. However, the rover has pitched upward, capturing a large portion of the sky. This violates the basic majority-foreground assumption, resulting in an output mask that includes the overcast sky and excludes the ground plane. The second sample is from the June/2008 trial on Lemon Creek Glacier during a near "white out" condition. Here the ground and sky share similar coloration, and are difficult to distinguish even for human observers. Under these conditions the adaptive histogram threshold strategy completely fails to remove the background region.

The first type of error can be avoided with a small algorithm variation. One learning-based obstacle detection method uses a heuristically-selected seed region as training data [61]. Using the idea of a seed region, the adaptive histogram threshold

(a) (b)

(c) (d)

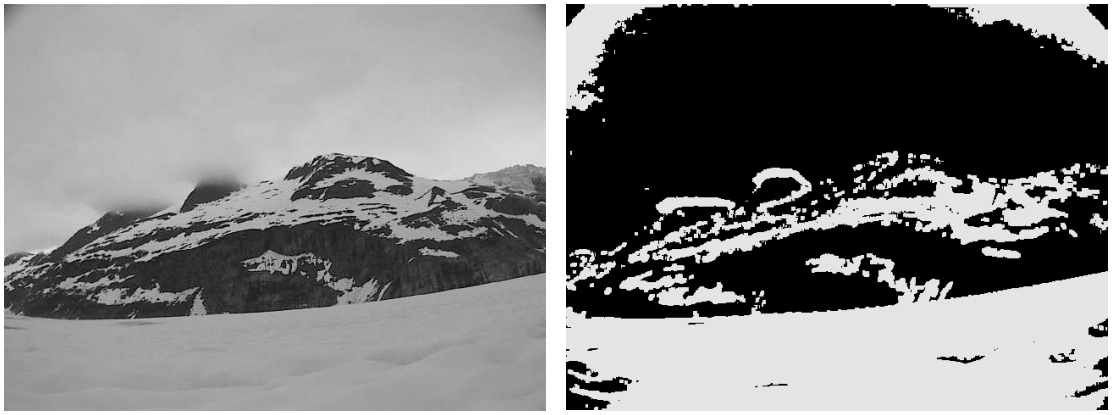
Figure 16: (a) A sample image from the June/2008 data set on Mendenhall Glacier, and (b) the mask produced by the adaptive histogram threshold method. This method incorrectly segments the sky, labeling the ground plane as background. (c) A sample glacial image from the June/2008 trial on Lemon Creek Glacier during a near "white out" condition, and (d) the resulting binary mask produced using the adaptive histogram threshold method. Under these conditions the adaptive histogram threshold strategy completely fails to remove the background region.

method has been modified to generate a grayscale histogram over a trapezoidal region directly in front of the rover. The histogram thresholds and final image mask are then generated as before. This has the advantage of ensuring that at least part of the ground in front of the rover will be included in the output region, eliminating the catastrophic failures present in the original algorithm. Further, as the histogram is calculated over a small part of the image, this method is also computationally faster than the original method. However, the issues of cloud inclusion remain unchanged. There is also a danger that the ground plane color variations will not be properly characterized, as a smaller area is used to calculate the threshold values. As such, the segmented regions tend to be smaller, improperly labeling ground plane areas as background. Figure 17 shows the output of the modified adaptive histogram threshold method on the same two sample images.

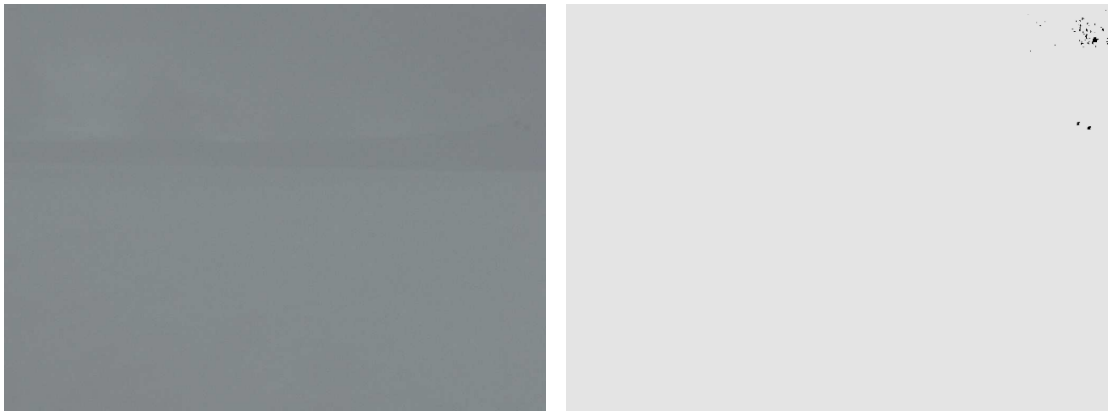
3.1.2 Region Growing

The MuCAR project at the University of the Bundeswehr Munich uses heuristic criteria to separate roadway pixels from the surroundings [75]. Similar color-based segmentation methods have been used successfully as elements of robotic ground plane segmentation systems [45, 90, 91]. Others find the addition of texture information to be useful in the segmentation process [100, 138, 140]. Region growing methods are common in segmentation algorithms, as they allow a large amount of freedom in the pixel test condition. Additionally, the output consists only of connected regions, an advantage over thresholding and other per-pixel evaluation methods.

A region growing scheme has been applied to the glacial foreground segmentation problem. A contextual region is defined by a trapezoid directly in front of the camera. The area within the contextual region is considered the seed region, R, and statistics are calculated in the form of mean and variance of the image intensity and image gradient. Relatively simple color and texture criteria have been implemented, shown



(a) (b)



(c) (d)

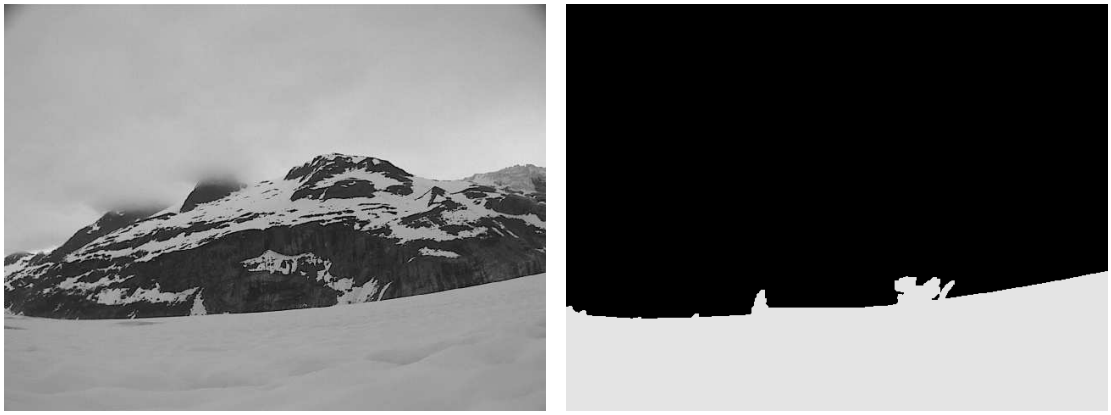
Figure 17: (a) A sample image from the June/2008 data set on Mendenhall Glacier, and (b) the resulting binary mask produced by calculating the adaptive histogram threshold on a seed region only. Use of a seed region allows the algorithm to correctly identify the ground plane color range, avoiding the catastrophic failure of the standard method. (c) A sample glacial image of a "white out" condition from the June/2008 trial on Lemon Creek Glacier, and (d) the associated foreground mask. This methods again completely fails to segment the background under difficult conditions.

in Equation (2). By only testing pixels adjacent to the current region, issues of cloud inclusion can be reduced. However, in instances where the cloud boundary contacts the ground plane, this method can still result in cloud regions being included in the final output mask. The computational complexity of this method depends on the calculation of the image gradient, a process known to be $O(n)$, and the region growing comparisons. As each pixel must be tested once, at most, the region growing phase is also $O(n)$.

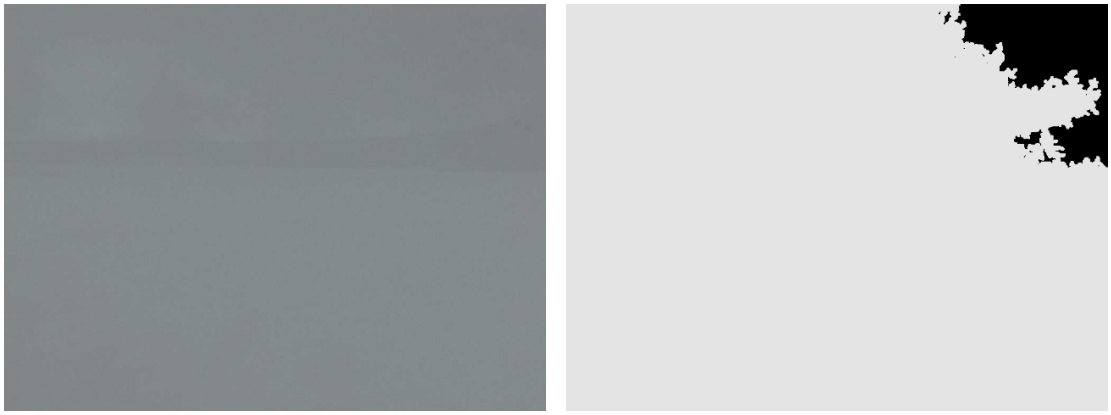
$$
\begin{aligned}
pixel(i,j) \in R \qquad & \text{if and only if} \\
& \mu_I - 2 \cdot \sigma_I < I(i,j) < \mu_I + 2 \cdot \sigma_I, \\
& \mu_G - 2 \cdot \sigma_G < G(i,j) < \mu_G + 2 \cdot \sigma_G, \qquad (2)
\end{aligned}
$$

where $I(\cdot)$ is the image intensity, and $G(\cdot)$ is the image gradient, μ_I, σ_I are the mean and standard deviation of the image intensity inside the region, and μ_G, σ_G are the mean and standard deviation of the image gradient inside the region.

Figure 18 shows the results of the region growing algorithm on the same example images. The prominence of the background mountains in the first image provides an unambiguous stopping boundary for the region growing phase. The connected region requirement also eliminates the sections of the overcast sky and snow-topped mountain peaks from the output. Further, since the algorithm was initialized with a foreground region, the output region is guaranteed to return the correct region. The catastrophic failures of the standard adaptive histogram threshold method cannot occur. In the second example image, a weak boundary exists between the ground plane and the overcast sky. As the region growing procedure examines each pixel individually, a single misclassification at the boundary can allow the region to expand into the cloud regions. Despite the inclusion of the clouds in the output region, this method still outperforms both variants of the adaptive histogram threshold.



(a) (b)



(c) (d)

Figure 18: (a) A sample image from the June/2008 data set on Mendenhall Glacier in which a large portion of the sky is included. (b) Since this region growing algorithm is initialized with a foreground seed region, the correct region is extracted. (c) A sample glacial image of a "white out" condition from the June/2008 trial on Lemon Creek Glacier, and (d) the mask produced by the region growing algorithm. Due to the weak boundary between the clouds and ground plane, a large section of the sky has been included in the output region.

While the region growing technique offers improved performance by utilizing texture information and guaranteeing a single, connected output region, it still uses only local information when evaluating each pixel. This can lead to the inclusion of undesired areas, such as clouds, if the boundary between the regions is weak. Statistical region merging (SRM), a variant of the region growing algorithm, partially solves this issue by using global information to select which pixel to evaluate next [90]. A function is used to rank the likelihood of neighboring pixels belonging to the same region. The most likely pairs are evaluated first, and merged into a single region if a certain color criterion is met. If the ranking function properly sorts the pixel pairs, then all pixel pairs within a single region will be tested before pairs that span regions. As a consequence, only over-merging errors can occur, when separate regions are incorrectly merged. A single tuning parameter is available that controls the level of inter-region merging. A simple ranking function has been implemented using the directional derivatives of the source image, shown in Equation (3). Assuming that true regions in the image are smooth, then large derivative boundaries only exist between regions.

$$\begin{aligned} cost(pixel(i,j), pixel(i+1,j)) &= \frac{\partial I(i,j)}{\partial x}, \\ cost(pixel(i,j), pixel(i,j+1)) &= \frac{\partial I(i,j)}{\partial y} \end{aligned} \tag{3}$$

where $I(\cdot)$ is the image intensity.

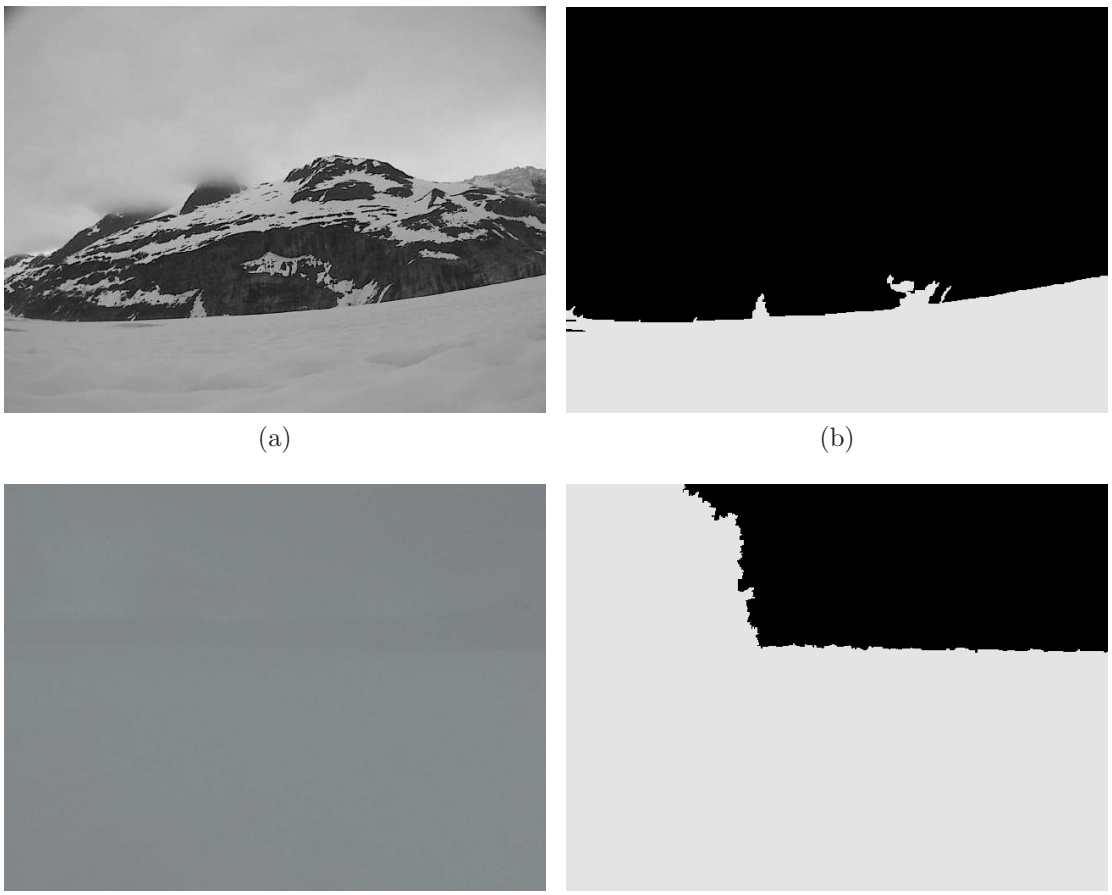
The SRM algorithm has been applied to the same set of sample images, the results of which are shown in Figure 19. Because the SRM algorithm uses global information to sort the pixel comparisons, separate regions for the sky, clouds, and ground are allowed to develop. This reduces the likelihood that sky and ground regions will be incorrectly merged. Despite this, the combination of a weak boundary, similar color ranges, and close proximity allowed some of the lower clouds in the upper left

of Figure 19c to be included in the ground plane region. It should be noted that this error region is significantly smaller than that produced by the standard region growing algorithm. Due to the prominent ground-mountain boundary, SRM performs similarly to region merging for the first sample image.

Due to the sorting and region tree generation, this method is the most computationally expensive of those examined. Given n pixels in an image, there are approximately $2n$ unique pixel neighbors in a 4-connected scheme. With efficient sorting algorithms, the computational complexity will be $O(2n \cdot log(2n))$. During the merging process, a hierarchical tree of region pixels is generated. Each of the $2n$ tests requires finding the parent pixel of both pixels being tested. Assuming a balanced tree structure, the parent look-up time will be $O(2n \cdot log(n))$.

3.1.3 Machine Learning

In the case of region growing, a merging criteria must be supplied, which ultimately determines the system performance. In the case considered, only grayscale intensity and gradient magnitude are considered. As the number of image attributes increases, it becomes increasing difficult to generate proper threshold values for the criteria equation. In contrast, the road segmentation method used by Stanley, the DARPA Grand Challenge entry from Stanford, uses a learned Gaussian mixture model to classify each image pixel by color [19], avoiding the need to set hard limits. Similarly, a water-sky segmentation system uses color and various texture measurements to successfully label each pixel, a problem visually similar to the glacial segmentation problem under consideration [37]. The same feature vector used by the water segmentation algorithm has been employed here, listed in Equations (4) - (10). Two Gaussian mixture models (GMM) [31] were trained on approximately 1% of the labeled images, consisting of nearly four million training examples. One GMM approximates the property distribution of foreground pixels, while the second was trained using background pixels.



(a) (b)

(c) (d)

Figure 19: (a) A sample image from the June/2008 data set on Mendenhall Glacier in which a large portion of the sky is included. (b) Similar to standard region growing, SRM produces an accurate region mask due to the strong ground-mountain boundary. (c) A sample glacial image from the June/2008 trial on Lemon Creek Glacier during a near "white out" condition, and (d) the resulting mask produced by the statistical region merging algorithm. By sorting the pixel comparison order, significant cloud and sky regions are able to develop, reducing the size of the incorrectly labeled portion of the image.

Each pixel in the test set is then classified based on the distribution with maximum likelihood membership.

$$\text{Intensity} \qquad I_{i,j} = I(i,j) \tag{4}$$

$$\text{Mean} \qquad \mu_{i,j} = \frac{1}{L} \sum_{i,j \in R} I(i,j) \tag{5}$$

$$\text{Standard Deviation} \qquad \sigma_{i,j} = \left(\frac{1}{L} \sum_{i,j \in R} \big(I(i,j) - \mu_{i,j} \big)^2 \right)^{\frac{1}{2}} \tag{6}$$

$$\text{Third Moment} \qquad \mu_{3_{i,j}} = \frac{1}{L} \sum_{i,j \in R} \big(I(i,j) - \mu_{i,j} \big)^3 \tag{7}$$

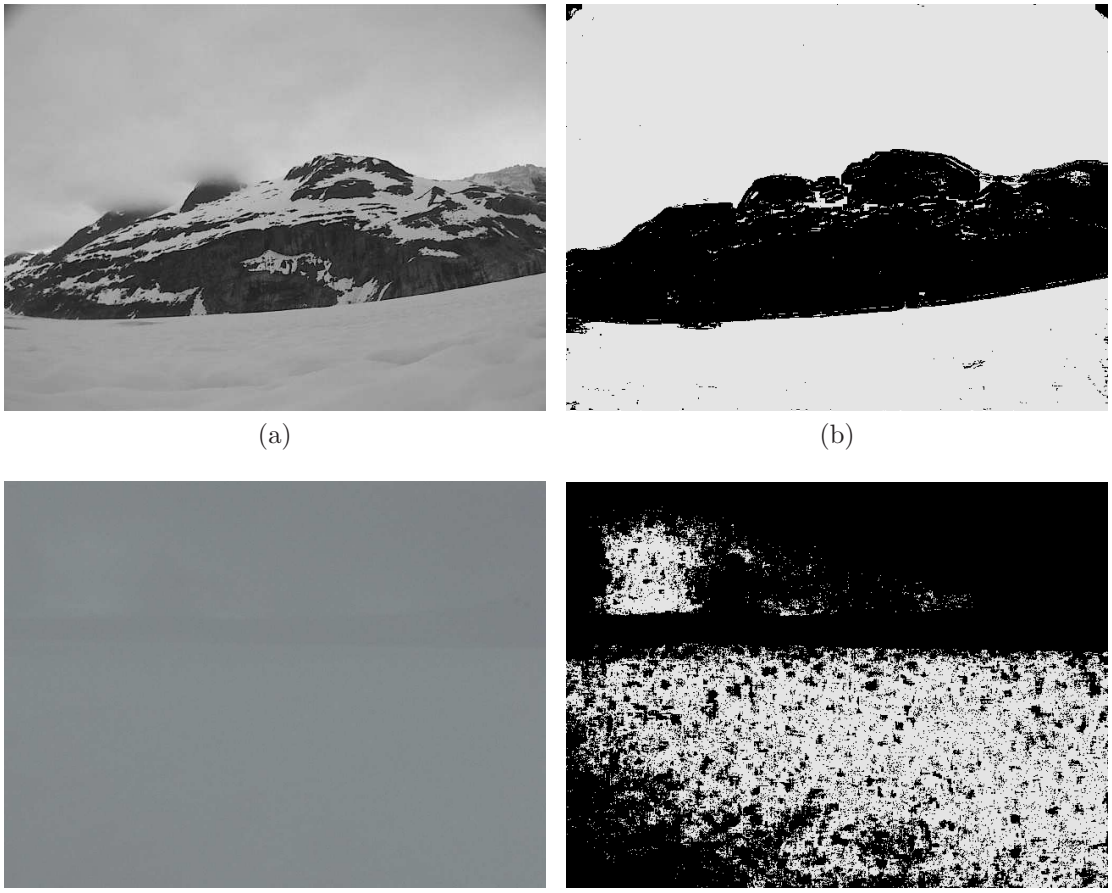
$$\text{Smoothness} \qquad S_{i,j} = 1 - \frac{1}{1 + \sigma_{i,j}} \tag{8}$$

$$\text{Uniformity} \qquad U_{i,j} = \sum_{i,j \in R} \overset{2}{\Pr} \left(I(i,j) \right) \tag{9}$$

$$\text{Entropy} \qquad e_{i,j} = - \sum_{i,j \in R} \Pr \left(I(i,j) \right) \log \big(\Pr \big(I(i,j) \big) \big) \tag{10}$$

where $I(\cdot)$ is the image intensity, L is the number of pixels in the small region, R, centered on pixel (i,j), and $\Pr\left(I(i,j)\right)$ is approximated using the histogram generated over region, R.

All the other methods previously discussed extracted a small number of properties from the image, but adapted the classification parameters based on those properties to the single image under consideration. This method, which uses a variety of image properties such as smoothness, uniformity, and entropy, to classify each pixel using a GMM, produces an unacceptably large classification error. This is illustrated in Figure 20, where the GMM classifier has been applied to the same two sample images. The issue with the GMM, or any training-based classification method, is that it must generalize its classification model across all the types of terrain encountered and all weather conditions. This results in a system that is typically less capable of discerning subtle differences between the ground and sky coloration properties.



(a) (b)

(c) (d)

Figure 20: (a) A sample image from the June/2008 data set on Mendenhall Glacier in which a large portion of the sky is included. (b) The mask obtained from the learning GMM. The foreground is correctly labeled, but large portions of the sky are also included. (c) A sample glacial image from the June/2008 trial on Lemon Creek Glacier during a near "white out" condition, and (d) the foreground classification results from the GMM. The mask includes only a small portion of the sky, but large portions of the ground plane have been removed.

To compute the output, the feature vector must first be generated. Each one of the vector components is computed from values in a small region, R, around the current pixel. This leads to a complexity of $O(L \cdot n)$, where L is the number of pixels inside R. As L will tend to be large, this dominates the complexity of evaluating the GMM.

3.1.4 Horizon Line Extraction

All of the previous methods mentioned use information local to the examined pixel to make segmentation decisions. However, the properties of glacial images make local examination problematic. Overcast skies, common in glacial environments, often share the same color range as the ground plane snow. Further complicating segmentation, the clouds and ground plane often intersect visually, making the determination of the horizon difficult. Figure 21 shows an example of this phenomenon in which a section of the ground-cloud boundary has been magnified. Using only the information within the magnified boxed, it is difficult, if not impossible, to find the true horizon line. When analyzing these images, humans tend to scan the image for visual cues in the form of strong horizon line segments. These line segments are then extended into image regions where the horizon is more ambiguous. Using this type of strategy, a ground segmentation method has been devised.

First, a set of color statistics are generated from a trapezoidal seed region, S, directly in front of the rover. By first constructing a pixel intensity histogram of this region, the median intensity, quartile values, and center 95% range can be efficiently calculated.

Strong line segments are then extracted from the image. An edge detector similar to the "Canny" operation [22] is used to find dominate image edges. Each edge is then simplified to be piecewise linear using the Ramer-Douglas-Peucker algorithm [52]. A

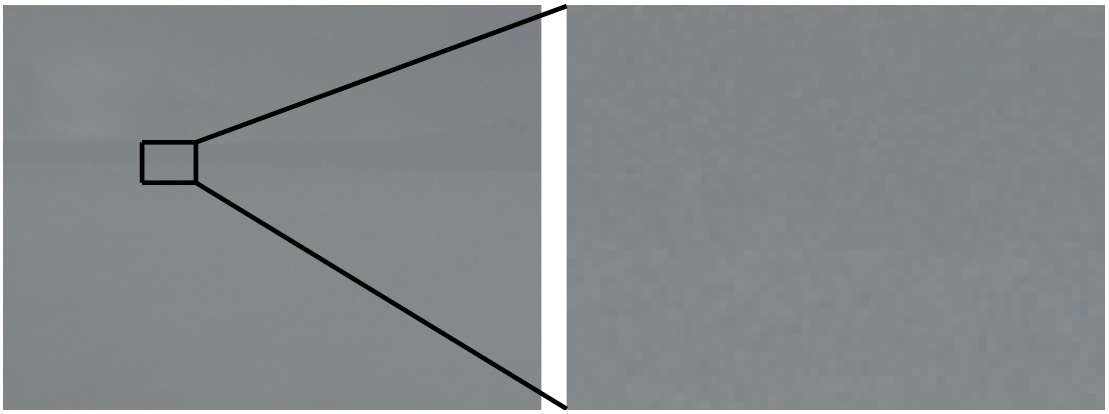


Figure 21: (a) A sample glacial image from the June/2008 data set with an enlarged region including a section of the horizon. The horizon line is easily distinguished in the whole image, but nearly invisible in the enlarged section.

minimum segment length constraint is enforced to remove the large number of noise-induced edges. Each remaining line segment is considered a candidate horizon line segment. A set of heuristic properties are then calculated for each candidate, designed to test the likelihood that the candidate is actually part of the true horizon line. These properties are summarized in the following.

Segment Length

Longer line segments are more likely to be part of larger structures, such as the horizon or mountain boundaries, and less likely to come from localized surface texture. Therefore longer line segments produce a larger weight, W_{len}, than shorter segments.

$$W_{len} = Length_{segment}/Width_{image} \tag{11}$$

where $Length_{segment}$ is the euclidean length of the candidate segment, and $Width_{image}$ is the image width in pixels.

Color Below Segment

If a segment is part of the horizon line, then the region below the line segment should be statistically similar to the foreground seed region. A pixel intensity histogram is constructed from an area, B, immediately below each candidate. Using

this histogram, the quartile intensity values are calculated and compared to the seed region statistics. The normalized euclidean distance between the quartile values of the seed region and the quartile values of the area below the line segment is used to create the segment weight, W_{below}.

$$W_{below} = 1 - \alpha \cdot \|Q_B - Q_S\| \tag{12}$$

where Q_i represents the vector of quartile boundary intensities, α is a normalization constant, S is the seed region area, and B is a small area below the current candidate segment.

Color Above Segment

If the segment is part of the horizon line, then the area above the line should be statistically different from the foreground seed region. In a similar fashion to the 'Color Below Segment' property, the euclidean distance between the quartile values of the seed region and the quartile values of the region, U, above the line segment is calculated. The property weight, W_{above}, is given by Equation (13).

$$W_{above} = \alpha \cdot \|Q_U - Q_S\| \tag{13}$$

where Q_i represents the vector of quartile boundary intensities, α is a normalization constant, S is the seed region area, and U is a small area above the current candidate segment.

Color Column

Line segments are often generated at the upper edge of snow-covered mountain peaks, or at cloud-sky boundaries. This results in snow-colored pixels directly below a line segment, even though a section of non-white pixels exists between the line segment and the ground. By weighting each line segment by the percent of white pixels between the segment and the bottom of the image, these types of segments may be removed. Equation (15) gives the calculation method for weight W_{column}.

$$A(i,j) = \begin{cases} 1 & \text{if } Q_S(1) < I(i,j) < Q_S(3) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$W_{column} = \frac{1}{L} \sum_{i=x_1}^{x_2} \sum_{j=y_{seg}(i)}^{Height_{image}} A(i,j) \tag{15}$$

where L is the total number of pixels below the candidate line segment, (x_1, y_1) and (x_2, y_2) are the end points of the current candidate line segment, the function $y_{seg}(x)$ returns the y-value of the candidate segment at a specific x-value, Q_S represents the vector of quartile boundary intensities of seed region, S, and $I(x, y)$ is the image intensity.

Distance From Predicted Horizon

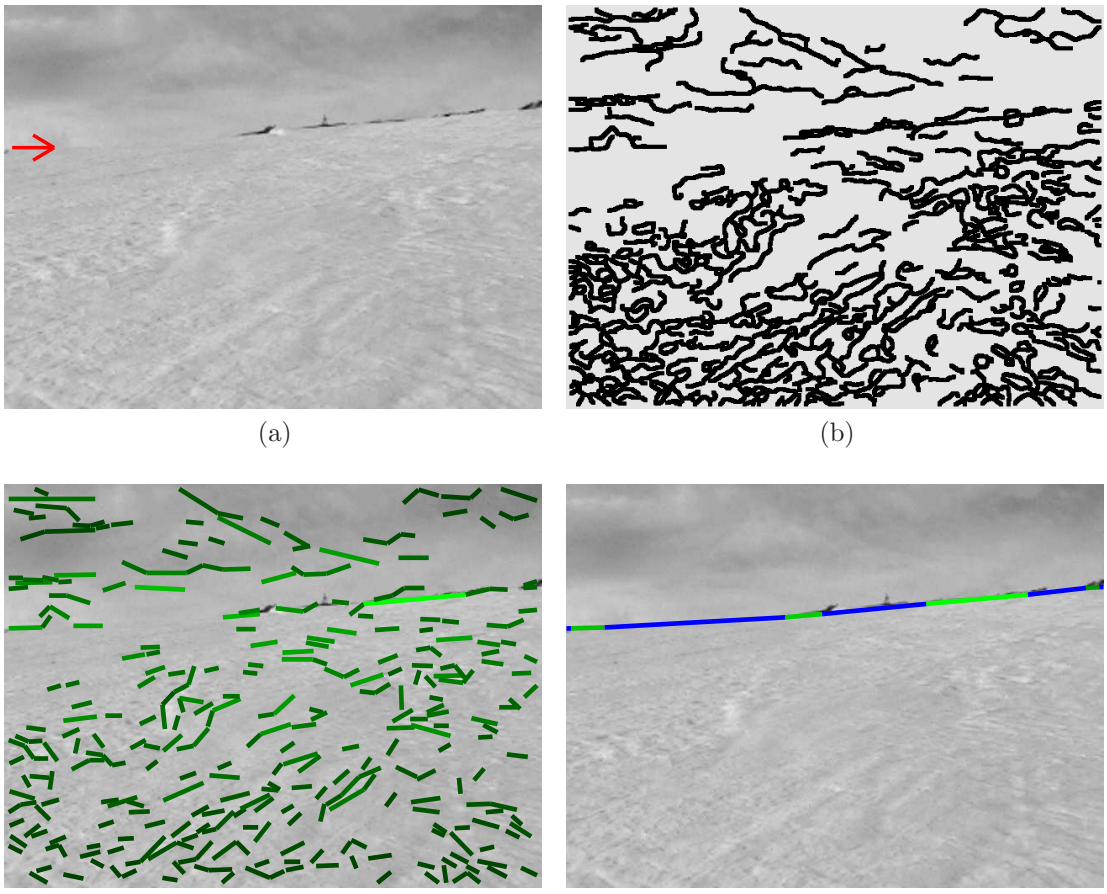
The position of the horizon location can be estimated using the current pose estimate of the camera from the odometry system, assuming a flat ground plane if additional topographic information is unknown. The distance from this estimate to a candidate line segment can be used as a measure of the likelihood the given candidate is part of the horizon. A Gaussian kernel centered on the horizon estimate is used as the weight W_{dist}, shown in Equation (16). The variance of the Gaussian kernel can be tailored to the expected uncertainty of the robot pose estimate and the expected variability of the terrain.

$$W_{dist} = \frac{1}{L} \sum_{i=x_1}^{x_2} G_\sigma(y_{est}(i) - y_{seg}(i)) \tag{16}$$

where the function $y_{seg}(x)$ returns the y-value of the candidate line segment at a specific x-value, $y_{est}(x)$ returns the y-value of the horizon estimate at a specific x-value, $G_\sigma(x)$ is the evaluation of a Gaussian kernel function with zero mean and variance, σ^2.

A combined weight is calculated for each candidate segment as the product of the individual weights described above. The top scoring candidate is selected as a seed segment for the horizon line. A greedy search is then conducted starting at each endpoint of the seed segment. The cost of connecting the seed segment to another candidate segment minus the weight of that line segment is compared to the cost of simply extending the seed segment along its current trajectory to the endpoint of the candidate segment. The lowest cost solution is then executed, and the process repeats until the edge of the image is reached. In this way, candidate line segments that exhibit weak visual cues serve to reinforce the path of stronger segments, while segments with strong visual cues have the ability to redirect the path of the horizon. Figure 22 illustrates the major steps in constructing the horizon.

Figure 23 shows examples of the produced horizon line under different conditions. Figures 23a and 23b show the results of the horizon line extraction process on the same two sample images used in the discussion of the other masking algorithms. In both cases the correct horizon was found, despite the presence of "white out" conditions in Figure 23b. Figures 23c and 23d illustrates a similar situation under different weather conditions. Figure 23d is from the June/2008 data set acquired at Lemon Creek Glacier under low lighting conditions, while Figure 23c is part of the May/2009 data set, which occurred under favorable weather at Site A on Mendenhall Glacier. In both images, a weak horizon line exists with snow-covered mountains immediately behind. This causes the horizon to visually blend with the background, which makes finding the correct horizon line difficult, even for human observers. Despite this, the horizon line extraction process is able to identify a reasonable horizon in both images. Figures 23e and 23f are images from the same data set, acquired several seconds apart. As the camera pans to the right, the horizon weakens to the point of becoming invisible. Figure 23f illustrates the point at which the desired horizon becomes too weak for successful detection. Despite the failure, the extraction



(a) (b) (c) (d)

Figure 22: (a) The sample image of Lemon Creek Glacier during overcast weather in June 2008. (b) The major edges extracted from the image using a "Canny" edge detection operator. (c) The extracted edges are approximated by piecewise linear segments, and weighted according to several heuristic cues. The intensity of the line segment color is proportional to the segment weight. (d) The highest weighted line has been used as a seed segment for the horizon line, connecting to other nearby line segments when possible. Original lines segments are shown in green, inferred sections are blue.

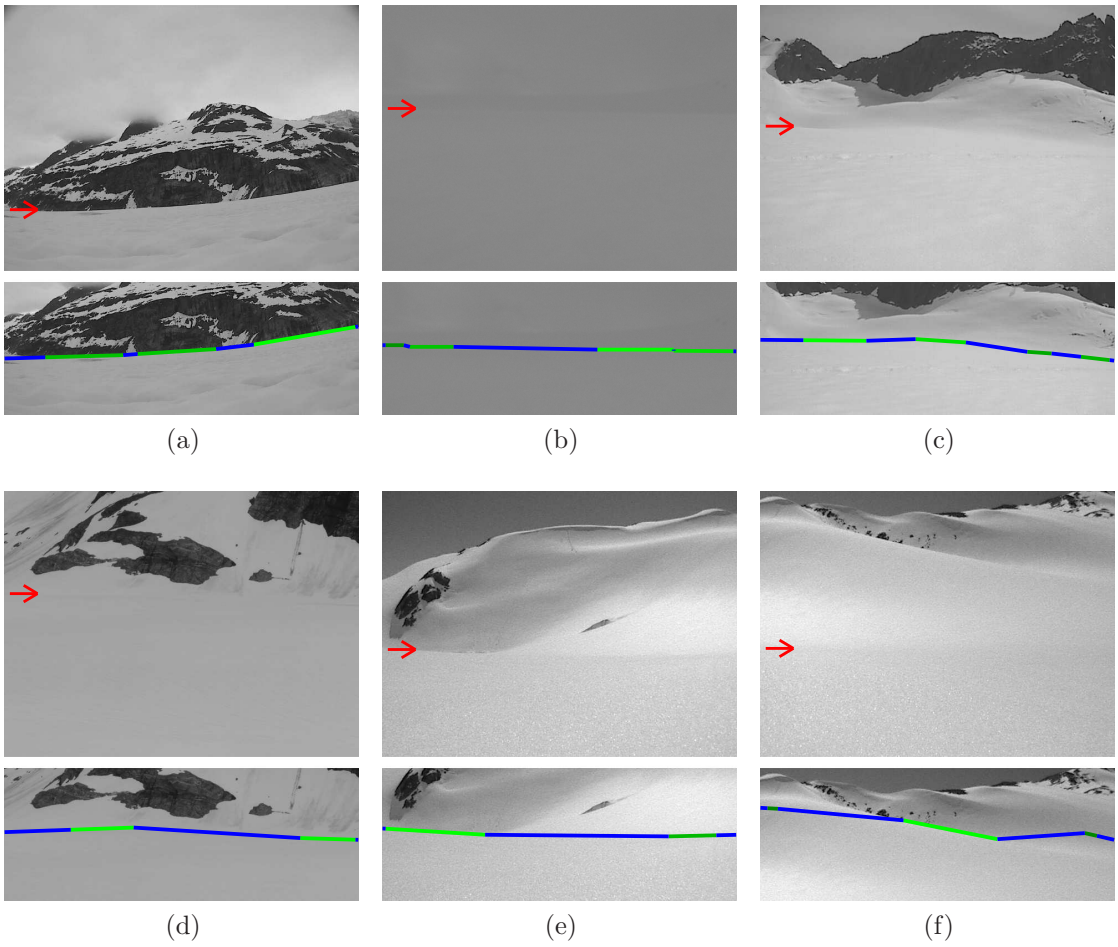


Figure 23: Typical results of the horizon line extraction process on images acquired on Mendenhall Glacier and Lemon Creek Glacier near Juneau, Alaska. Top graphic shows the original image, while the bottom shows a truncated section with the horizon line drawn. The desired horizon is indicated on the original image.

performance degraded gracefully, reverting to the next, stronger boundary line.

From a computation standpoint, most of the operations are applied per-segment, not per-pixel. This means the computation time is proportional to the number of candidate line segments, not the image size. Further, most of the operations require only a histogram of a small area, which is an inexpensive calculation. The resulting algorithm is capable of running in real time, with computation times of less than 30ms on a 640x480 image.

3.1.5 Results

To evaluate the effectiveness of the proposed region extraction algorithm, continuous segments of recorded video from each of the field trials were selected. The horizon line extraction algorithm (HL) was applied to each video segment, and the resulting region mask was recorded. To compare the results, images from each video segment were hand labeled, indicating the area of traversable foreground. It should be noted that in some images the line between traversable foreground and background is somewhat arbitrary. An effort was made to choose a consistent line between sequential images. Ultimately, this ambiguity exists only over a small vertical range of pixels in the image, and should not unduly affect the measured performance. Due to the manpower required to hand label images, only 100 frames were selected from each video segment, uniformly spaced through time. The algorithm results are then compared to the hand labeled images, with the number of incorrectly labeled pixels counted for each frame. For comparison, the methods of adaptive histogram thresholding (AHT), the AHT calculated on only a seed region (AHT-S), region growing (RG), Gaussian mixture model (GMM), and statistical region merging (SRM), were also evaluated in the same manner. The SRM algorithm is a more advanced region growing operation that employs global information to produce more accurate boundary locations [90]. However, the use of global information entails performing a per-pixel sort, a numerically expensive operation. The results of each test location are summarized in Figure 24 in the form of boxplots. Boxplots are a convenient graphical method of comparing statistical results that may not be normally distributed [77]. The box center-line indicates the median score, while the upper and lower border indicate the first and third quartile boundaries.

As illustrated in Figure 24, AHT and GMM consistently score the worst of all the methods investigated. These methods analyze each pixel without any consideration for the pixel location. Consequently, large portions of the sky tend to be misclassified.

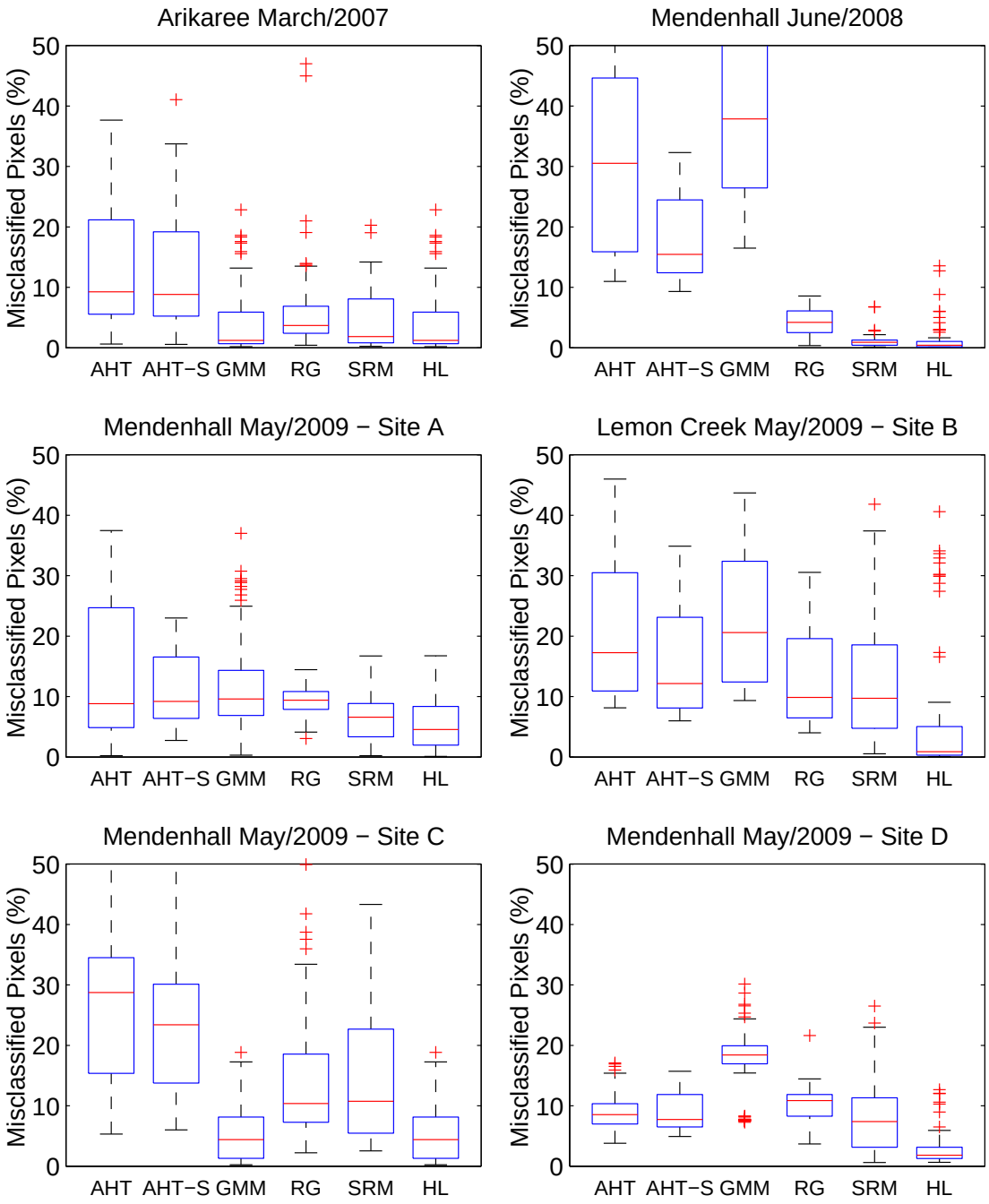


Figure 24: Classification performance results for adaptive histogram thresholding (AHT), modified histogram thresholding operating on a seed region (AHT-S), region growing (RG), statistical region merging (SRM), machine learning using a support vector machine classifier (ML), and the proposed horizon line extraction process (HL). Each algorithm was tested against 100 hand-labeled images from each of 8 different field test locations.

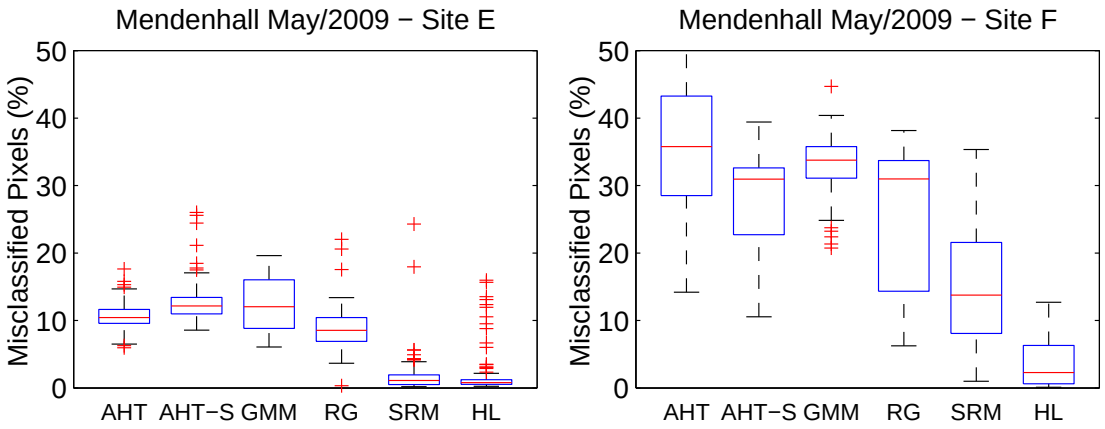


Figure 24: (continued)

In contrast, RG and SRM only consider pixels neighboring the current region. Since these algorithms were initialized with a foreground region, they are more likely to stay confined to the foreground, resulting in better segmentation performance. Additionally, SRM uses global information when selecting the next candidate pixel to merge, explaining the improved performance over the standard RG algorithm.

To test the real world performance of each algorithm as implemented, a single 640x480 test image was loaded into memory. Each algorithm then processed the test image 1000 times, and the elapsed time of the processing was recorded. For reference, these time trials were performed on an Intel Core Duo^{TM}T2500 running at 2.0 GHz.

The execution times of each algorithm are summarized in Table 2. The adaptive histogram threshold method was the fastest, followed by region growing. The statistical region merging method, which consistently exhibited high accuracy, was only capable of operating at 3 to 4 Hz. Such low frame rate operation is unacceptable as a component of a real-time obstacle avoidance system. The Gaussian mixture model was also incapable of real-time operation. While the classification stage itself was sufficiently fast, the generation of the required feature vector was too computationally intensive. Finally, the horizon line extraction procedure was capable of greater than 20 Hz operation, making it an acceptable choice.

Table 2: Region Of Interest Algorithm Execution Times

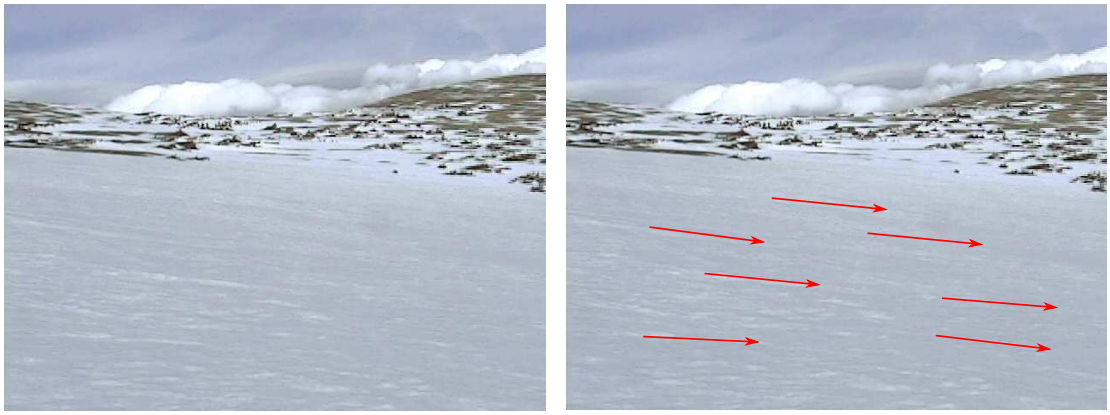
Algorithm	Execution Performance (Average Time Per Frame)
Adaptive Histogram Threshold	0.0074 s
Adaptive Histogram Threshold (Seed Region Only)	0.0043 s
Region Growing	0.0229 s
Gaussian Mixture Model	0.2677 s
Statistical Region Merging	0.2803 s
Horizon Line Extraction	0.0296 s

3.1.6 Conclusions

A custom horizon line extraction algorithm based on visual cues was proposed. Due to the real-time, low processing requirements of field mobile robotics, special emphasis has been given to the time complexity of each algorithm. The performance of each algorithm is evaluated numerically, both in terms of time and accuracy, on samples from each of ten different field trials on glaciers in Alaska.

From a classification performance standpoint, the proposed horizon line extraction procedure (HL) and statistical region merging (SRM) are the clear winners. In all conducted trials, the median classification error of the HL algorithm was less than 2.5% of image pixels. This result marginally outperformed statistical region merging (SRM) in every data set, but does not require the computational expense of a full pixel sort. In terms of execution speed, the HL algorithm is an order of magnitude faster than SRM, enabling real-time operation.

Further, this method has been formulated to avoid the use of as many thresholds and tuning parameters as possible. Each visual cue is evaluated for each candidate line segment; it is not until the end of the process that candidates get culled. If the properties to which a specific visual cue responds are not present in the image,



(a) (b)

Figure 25: (a) A sample glacial image from the March/2007 data set on Arikaree Glacier, Colorado. (b) Subtle surface texture gives visual cues as to the terrain slope, as indicated.

then all candidate segments get penalized a similar amount. In contrast, a threshold-based system would likely cull all of the candidates, resulting in failure. This endows the system with a certain robustness to image variation, and mitigates the effects of applying visual cues that are not optimal for a specific image.

3.2 Visual Slope Estimation

A human can quickly and accurately estimate the slope of the environment, such as that shown in Figure 25. From this single image, without any depth information, one can successfully estimate slope and loosely define distances from the camera. Our ability to estimate the slope comes from visual cues in the form of directional texture features that align with the perceived slope. We exploit these features to create a terrain slope estimation process using only single camera imagery [127, 128].

3.2.1 Preprocessing

The image foreground is isolated using the horizon line extraction process described in Section 3.1. This limits the processing to the ground plane, the area of consequence for ground-based mobile robots. In order to extract reliable information from the

directional texture cues, the surface texture must first be enhanced. The adaptive contrast enhancement method described in the following is applied for this purpose.

To enhance the contrast, a mapping, $f(i)$, is defined between the input image intensity and the output intensity. The contrast of a specific intensity level, $C(i)$, is then defined as the rate of change of output intensity level, as shown in Equation (17).

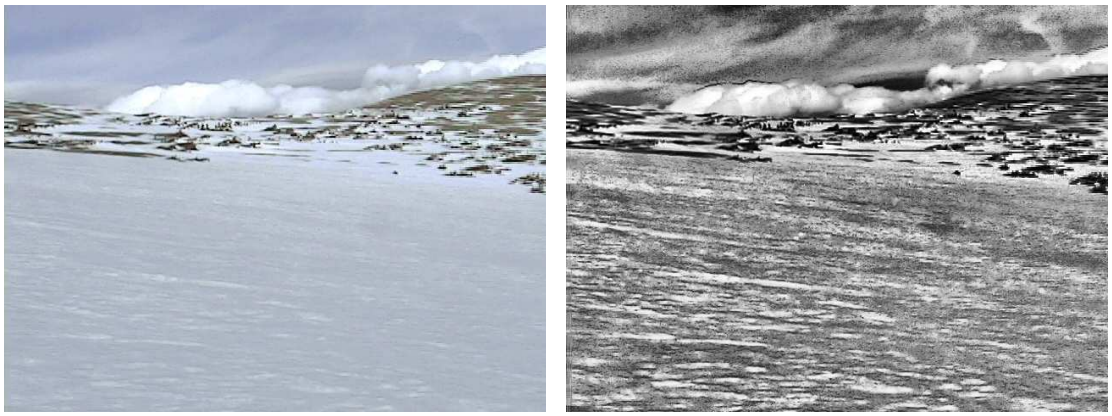
$$C(i) = \frac{\partial f}{\partial \tau}|_{\tau=i} \approx f(i) - f(i-1) \tag{17}$$

Since the output image range is usually defined to be the same as the input intensity range, the contrast of a specific intensity value can only be enhanced at the expense of others. Adaptive histogram equalization (AHE) is a standard nonlinear contrast enhancement method that uses the histogram of the image to define the mapping function, as shown in Equation (18). This interesting choice of mapping function applies the greatest amount of enhancement to the most common intensity values. In terms of arctic images, the most common intensity values will be near-white, which is exactly the range in which increased contrast is desired.

$$f_{ahe}(i) = \alpha \int_0^i histogram(\tau) d\tau \tag{18}$$

where α is a normalization factor generally set to the image pixel count, and the function $histogram(\tau)$ returns the number of pixels in the image with intensity τ.

A variant of adaptive histogram equalization has been formulated to enhance x-ray images and CT scans [98], which imposes a contrast limit on the mapping function. Contrast limited adaptive histogram equalization (CLAHE) separates the image into different contextual regions. Within each region, a histogram equalization procedure is calculated. To prevent over-enhancement of local areas, a contrast limit is imposed. This applies an upper bound to the slope of the mapping function, resulting in smoothly varying contrast. The resulting mapping functions are shown in Equations (19) and (20).



(a) (b)

Figure 26: (a) A sample glacial image from the March/2007 image set, and (b) the results of the CLAHE image enhancement. After processing, the underlying scene structure is clearly visible.

$$C_{clahe}(i) = \begin{cases} histogram(i), & histogram(i) < C_{max} \\ C_{max}, & histogram(i) \geq C_{max} \end{cases} \tag{19}$$

$$f_{clahe}(i) = \alpha \int_0^i C_{clahe}(\tau) d\tau \tag{20}$$

where the function $histogram(i)$ returns the number of pixels in the image with intensity i, C_{max} is the clip limit, $alpha$ is a normalization constant generally set to the image pixel count, and f_{clahe} is the resulting mapping from input to output intensity values.

Additionally, this method has a single tuning parameter, the contrast limit C_{max}. A good setting for this value was determined experimentally once, and used in all subsequent images and field tests. Figure 26 shows the results of this contrast enhancement on an example image from the March/2007 data set. After processing, the underlying scene structure is clearly visible.

3.2.2 Sparse Slope Estimates

A set of sparse terrain slopes can be estimated using a Hough transform [128]. The Hough transform calculates every possible line to which a specific edge pixel could

belong [30]. These lines are generally represented by a (ρ, θ) pair, where ρ is defined as the shortest distance between the origin and the line, and θ is the angle between the line and the X-axis. Using these definitions, the functional relationship between ρ and θ shown in Equations (21) and (22) can be derived for a specific image pixel, (u, v). The relationship between the ρ-θ representation and the more common slope-intercept form can be seen in Equation (21).

$$v = \left(-\frac{\cos\theta}{\sin\theta}\right) u + \left(\frac{\rho}{\sin\theta}\right) \tag{21}$$

$$\rho(\theta) = u\cos\theta + v\sin\theta \tag{22}$$

Dominant texture edges are extracted using a Canny edge detector [21]. The Hough transform then converts each edge pixel in the image space, (u, v), into a sinusoidal line in the ρ-θ parameter space. As each image pixel is transformed, the sinusoids in the parameter domain will tend to intersect if the image pixels are in a straight line. The number of sinusoids that intersect in a particular location is an indication of the strength or confidence of the line. Thus, the (ρ, θ) pair corresponding to the maximum confidence values in the parameter space may be selected as being representative of that region. Figure 27 shows the resulting Hough-space representation of the preprocessed image, which shows specific areas of high intensity corresponding to common directionality of surface texture.

When a pixel is transformed into the parameter space, it loses any sense of its location in the original image. It is therefore common for many local maxima to occur in a very small neighborhood. This results in having slope data only for a small area in the total region of interest. To overcome this problem, the image is divided into smaller subimages, and the Hough transform is applied separately to each subimage. In this way, the extracted slope can be applied to a specific area of the original image, and slope data will exist for all areas in the image. In each subarea, only one slope

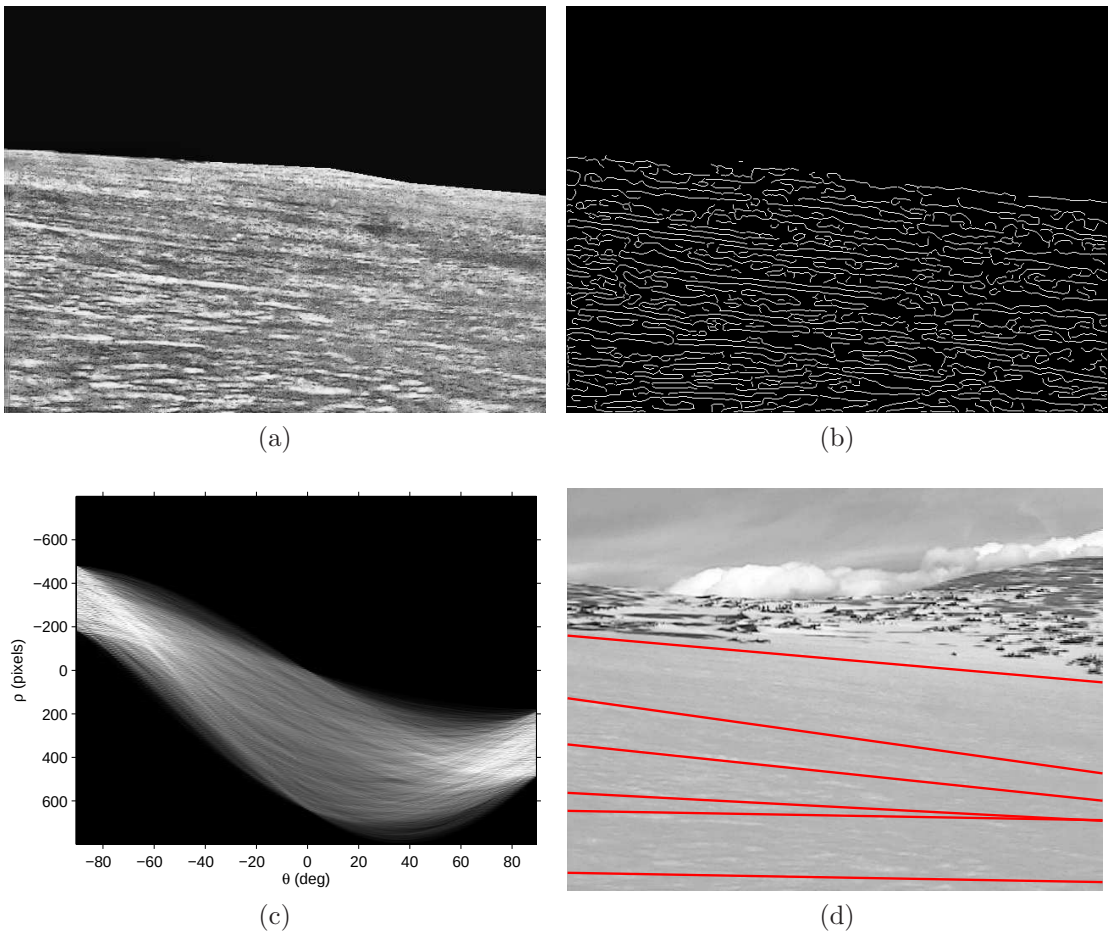
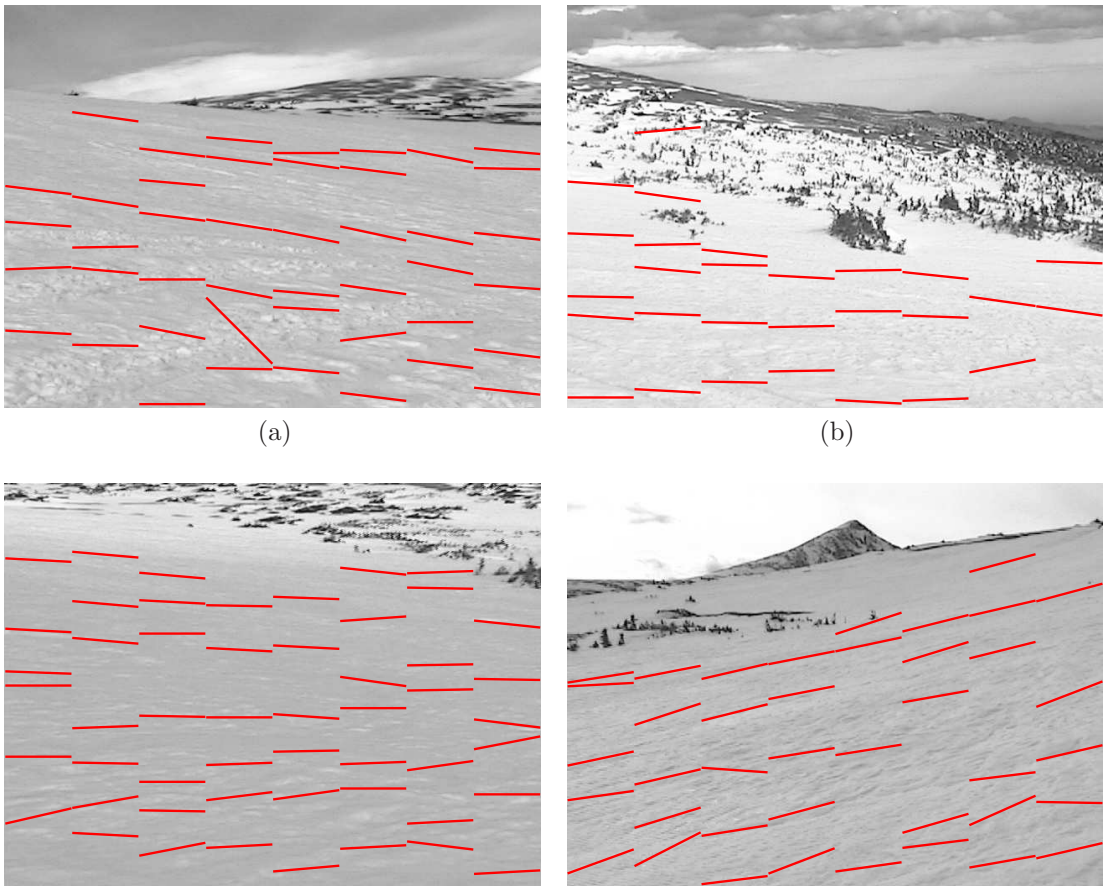


Figure 27: (a) A sample glacial image from the March/2007 image set after masking and contrast enhancement preprocessing steps. (b) The sample image after applying the Canny edge detector. Dominant texture elements have been preserved in the edge image. (c) The resulting ρ-θ image after applying the Hough transformation. (d) Several local maxima of the Hough transform have been extracted and overlaid on the original image. The Hough lines approximate the directionality of the surface texture.

is desired, lending itself to the Fast Hough Transform algorithm [69], which employs integer shifts instead of floating point operations to reduce the processing time.

Figure 28 shows the sparse slope estimate results on a representative sample from the March/2007 data set. The images in Figures 28a and 28d exhibit the most prominent slope characteristics. In both cases, the sparse estimates generally align with the expected direction, slanting down and right in the case of Figure 28a and more aggressively down and left in the case of Figure 28d. Because this method estimates a single slope for an entire image block, image structures that were not removed by the horizon mask can interfere with the slope estimate. This is most clearly seen in the grassy areas in the upper left of Figure 28b. Additionally, the bottom-center areas of Figures 28a and 28b have areas of disturbed snow. This was caused by snowmobile vehicles used to travel to the test site. This disruption in the natural surface texture leads to incorrectly steep slope estimates. However, due to the remote deployment locations, the rovers would rarely encounter such a disrupted snow surface in practice.

The slope estimates obtained for the previous glacial images could only be evaluated qualitatively, as the ground truth information could not be obtained. To obtain numerical results, an early version of the simulation system described in Chapter 2.3 was utilized. Because the slope estimates generated by this method are two dimensional in nature, special care has been taken when exacting comparable data from the simulation system. When the rover's camera views the landscape, the three dimensional terrain is projected onto a two dimensional plane perpendicular to the camera's line of sight. In order to generate ground truth for the simulation that would be comparable, a similar approach was followed. For each slope estimate, a ray was projected from the camera to a terrain patch at the center of the slope estimate line. Once the intersection of this ray and the terrain is determined, the elevation of the terrain on either side of the intersection point is measured. The ground truth



(a) (b)

(c) (d)

Figure 28: Sparse slope estimates performed on a variety of images from the March/2007 data set.

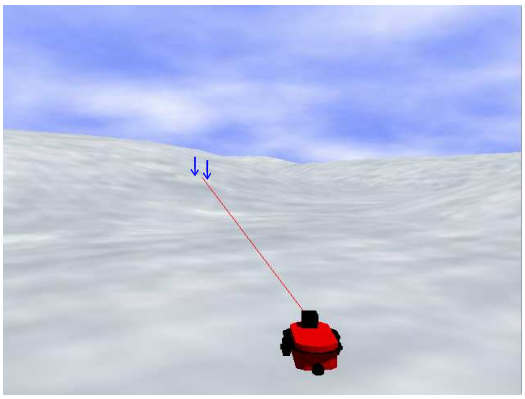


Figure 29: Visual slope estimates versus the ground truth data obtained from the simulation environment.

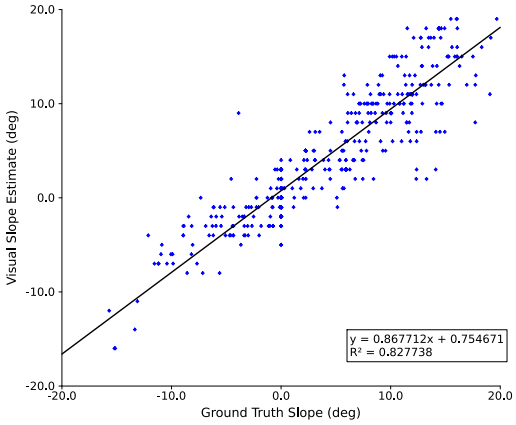


Figure 30: Sparse visual slope estimates versus the ground truth data obtained from the simulation environment.

slope is then calculated from the two measured elevations. Figure 29 illustrates this process.

With the data collection method in place, the rover was driven manually through a section of the simulated terrain while the generated visual slope estimates and ground truth data were logged approximately once per second. Figure 30 shows a comparison between the ground truth data and the visual slope estimates obtained during a 60 second traverse. As can be seen, the visual slope estimates are highly correlated with the ground truth data, with a correlation factor above 0.9. The best fit line has a slope of 0.86, indicating this method tends to mildly underestimate the larger slopes. Over the data set presented, the error between the estimate and ground truth value exhibits a near-zero mean, with a standard deviation of less than 3.0 degrees. The estimates therefore provide a good indication of the terrain slope.

3.2.3 Dense Slope Estimates

While the numerical accuracy of the slope estimate system presented in the previous section is promising for use as input into a control system, the sparse nature of these estimates is problematic. The sparse slope estimate method relies on detecting

medium scale surface features within the image. However, these visual features are not evenly distributed through the entire image, leading to large areas without good estimates. Further, a single feature measurement was used to represent a large image region, allowing an outlier image structure to unduly impact the result. The following method attempts to remedy these shortcomings, while additionally providing dense slope estimates [129].

The contrast-enhanced surface texture exhibits a desirable slope-alignment property, but alignment noise is a larger issue when dealing with small-scale texture features. Similar to the area of fingerprint enhancement [53, 60] where it is desired to find and follow the small ridge details of a print, a ridge orientation can be defined not by the direction of a single texture element, but rather by the common direction shared by a small region around each pixel. The slope estimate is produced by finding the least square estimate of the dominant Fourier spectrum direction within a small neighborhood.

To calculate the orientation of a given pixel, (u, v), the image gradient within a neighborhood of that pixel is first calculated. Then the two component vectors, $\vec{\nu_x}$ and $\vec{\nu_y}$, are generated, as described in Equations (23) and (24). The orientation, θ, is then defined as the least squares solution to Equation (25). The entire slope calculation process can be calculated in real-time.

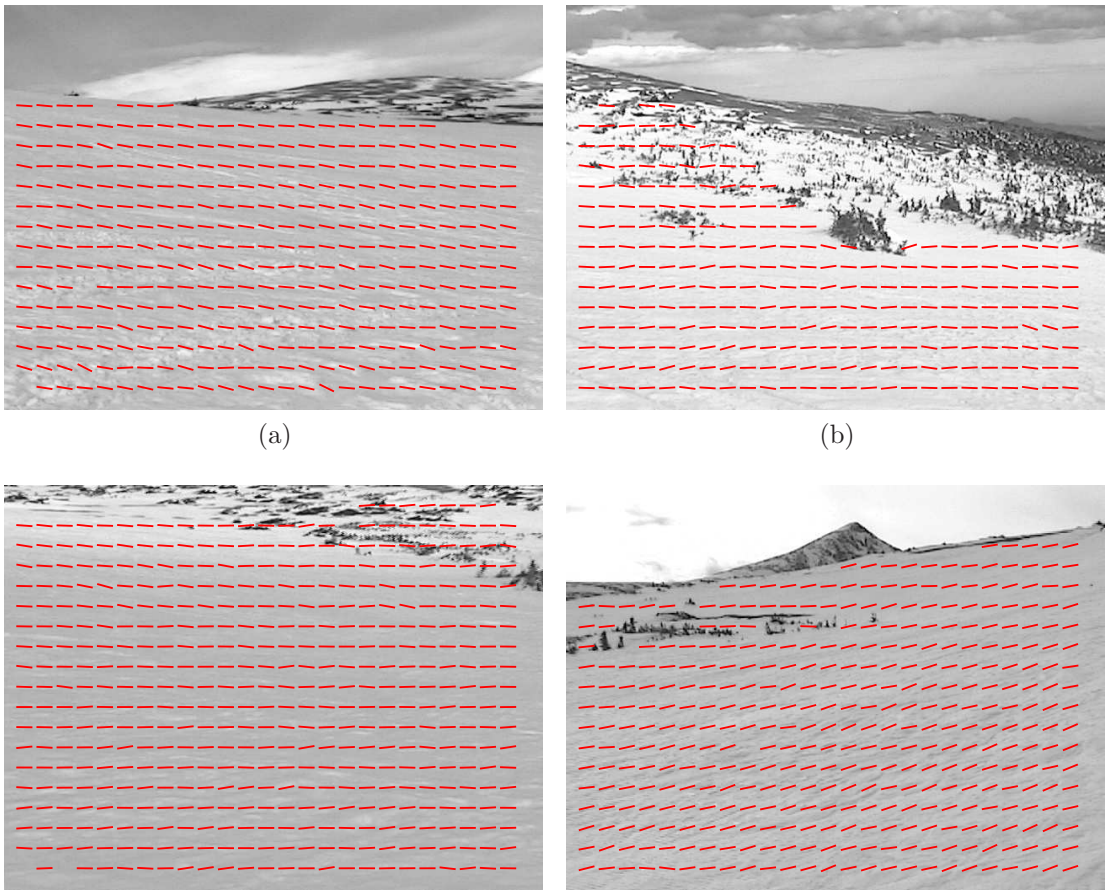
$$\vec{\nu_x}(u,v) = \sum_{neighborhood} 2\partial_x(u',v')\partial_y(u',v') \tag{23}$$

$$\vec{\nu_y}(u,v) = \sum_{neighborhood} \partial_x^2(u',v') - \partial_y^2(u',v') \tag{24}$$

$$\theta(u,v) = \frac{1}{2}\tan^{-1}\left(\frac{\vec{\nu_y}(u,v)}{\vec{\nu_x}(u,v)}\right) \tag{25}$$

where (u', v') are pixel coordinates within the neighborhood of (u, v).

For comparison, the dense slope estimation algorithm has been applied to the



(a) (b)

(c) (d)

Figure 31: Dense slope estimates extracted from a variety of images from the March/2007 data set.

same images from the March/2007 data set used in Section 3.2.2, with the resulting images shown in Figure 31. As before Figures 31a and 31d, which have strong texture properties, produce visually consistent results across the entire image. The dense method is also better able to characterize the foreground section of disrupted snow in Figure 31b. While the sparse method is only able to provide a single data point in this area, the dense slope information indicates boundaries of the disturbed snow with downward slanting slope values.

Additional examples of processed glacial terrain are provided in Figure 32. In the first image set from Lemon Creek Glacier, the terrain grade in the original image is virtually invisible. Yet, the dense estimate process is able to provide reasonable

results, even in the areas that originally seemed uniformly white. The second pair of images illustrates a large crevasse on Mendenhall Glacier. The slope estimation process is able to handle both the snow and exposed ice textures without modification, as well as handle the quickly varying terrain grade. The estimates provided clearly show the snow and ice sloping into the mouth of the crevasse, while a relatively safe area exists in the far left. Since the sparse method operates on large image blocks, it would be unable to handle the complexity of this terrain.

As before, the dense slope estimates can only be evaluated qualitatively on images collected from the field campaigns, as no source of ground truth terrain data on an appropriate scale is available. Instead, the simulation system presented in Chapter 2.3 has been utilized for numerical evaluation. Before using the simulation to validate a specific visual algorithm, the rendering quality of the simulation system must be investigated. For the dense slope estimation system, the main visual property of interest is related to the directionality and strength of small-scale texture elements. A sequence of 300 frames from Site D have been compared to 300 frames of the simulation, rendered using the same control commands provided to the real rover. The image gradient has been calculated for each frame, and the distribution of gradient magnitudes and gradient directions have been collected for each set. Figure 33 shows the comparison of the gradient scores of the real image sequence and the corresponding simulation image sequence. While the medians of the two data sets are similar, the range of the data sets differs significantly. Within the simulation system, the major terrain coloration is derived from a single texture image. As outdoor simulations tend to be large, a single pixel from this texture could span several meters. To compensate for this, a second detail texture is tiled across the surface and blended with the main texture. It is the detail texture that is responsible for the local-scale details present in the simulation. However, due to this tiling procedure, the camera tends to see the same detail repeatedly, lowering the structural variance of the simulation over a large

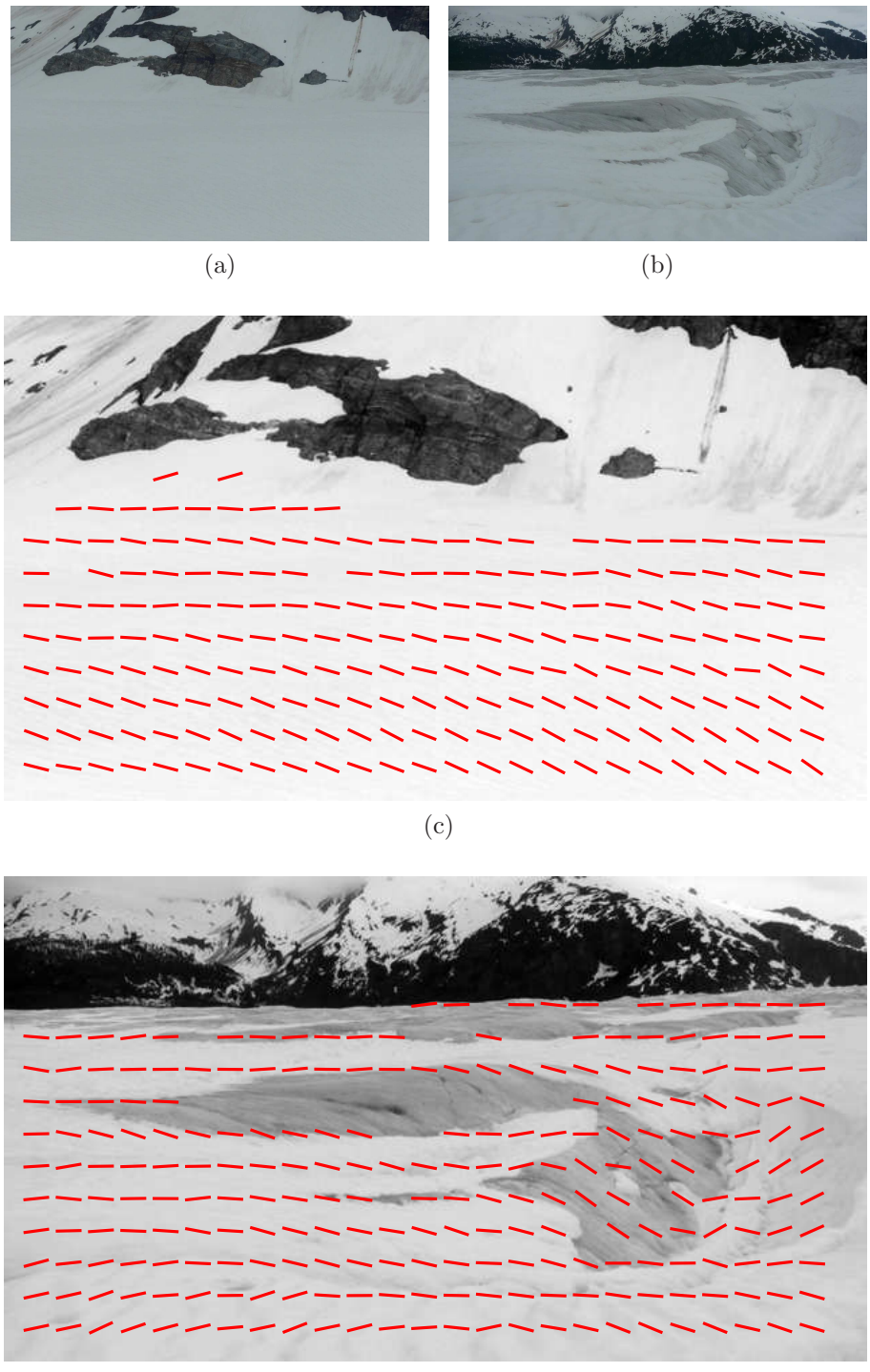
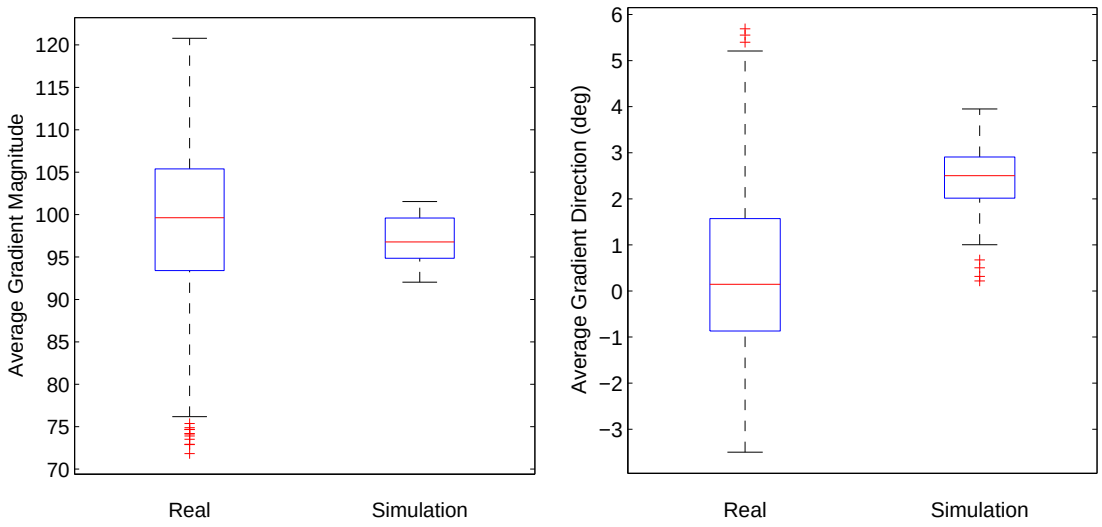


Figure 32: Images (a) and (c) show an image from Lemon Creek glacier with nearly invisible surface texture, and the processed slope estimates, respectively. Images (b) and (d) show a large crevasse at Mendenhall glacier and the resulting slope estimates. The slope profiles in (d) clearly show the elevation changes at the edge of the crevasse.



(a) Gradient Magnitude (b) Gradient Direction

Figure 33: Comparison of performance metric statistics for the dense slope estimate algorithm. As the main visual property of interest is related to the directionality and strength of small-scale texture elements, image gradient metrics of average magnitude and direction.

number of frames. Additionally, factors such as lighting changes and camera noise are present in the real data, further increasing the variance of the results over time.

With an understanding of the simulation quality, the accuracy of the dense slope estimates can be investigated. 100 frames were rendered at random locations and yaw orientations within the simulation system. The dense slope estimate algorithm was applied to each frame, and the resulting slopes stored. The true terrain slope was extracted from the simulation at the center of each pixel. The visual slope estimates were then fit to a linear model of the form $y = a \cdot x + b$. A scatter plot comparing the estimated slopes to the ground truth slopes is provided in Figure 34. From the regression analysis, almost 80% of the variability can be explained by the linear model. The correlation coefficient for the dense slope estimate system is only marginally lower than that of the sparse estimates, while providing significantly more terrain information. Additionally, the dense estimate system generates lower average errors, with an error standard deviation of 2.3° versus 2.9° for the sparse system.

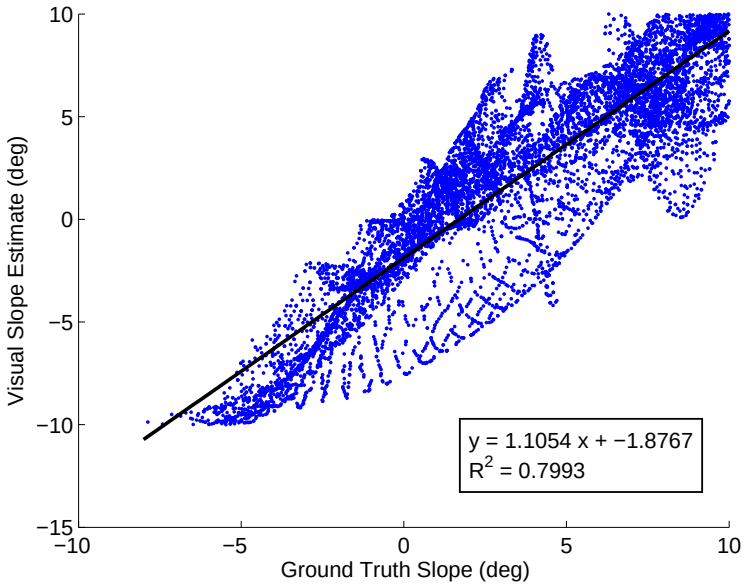


Figure 34: Dense visual slope estimates versus the ground truth data obtained from the simulation environment.

3.2.4 Conclusions

Due to the nature of glacial terrain, estimating the terrain slope is a key aspect in determining the traversability of the environment and the detection of navigation obstacles. Although direct measurement of the terrain via a ranging sensor may be ideal for these purposes, these types of sensors tend to be expensive, have high power requirements, and can be confused by the specularity of the environment. Instead, two vision-based methods have been developed that exploit the presence of natural erosion texture of the surface to perform a terrain slope estimate.

Both methods have been tested qualitatively against images from glacial field trials, and quantitatively inside a 3D glacial simulation system. While both methods perform well inside the simulation system, with linear correlation coefficients above 0.9, the dense slope estimate method provides significantly more information to the control system, and is able to capture smaller scale terrain features. As both methods are also capable of running in real-time, the dense slope estimate system has been selected for use in subsequent control schemes and terrain modeling data needs.

3.3 Slope-based Control

In the previous section, it was shown in simulation that the generated visual slope estimates are highly correlated to the true terrain slope. However, real terrain exists in three dimensions; formations such as ridge lines and saddle points present uniquely different slopes based on the orientation of the observer. However, the visual slope estimate system only approximates the terrain slope in a single plane. To demonstrate that these single-plane slope estimates are still useful as input to a controller of a ground-based mobile robot, a behavior-based reactive control scheme has been implemented with the goal of minimizing the chassis roll experienced by the rover.

A version of the distributed architecture for mobile navigation (DAMN) [99] has been implemented as the robot control scheme. This system is an example of the more general class of voting-based control techniques in which different, and possibly competing, objectives each vote for potential control outputs. This allows the seamless integration of multiple behaviors, the combination of reactive and deliberative behaviors, as well as behaviors that update at different time intervals, without requiring each behavior to even be aware that other behaviors exist within the system.

The DAMN architecture focuses more heavily on the steering control, placing a supporting role on velocity control. Each behavior in the system weights each possible steering control based on its own situational assessment. For example, an obstacle control strategy might weight "turn left" and "turn right" controls highly when approaching an obstacle, while a path planner might weight the steering controls that direct the robot away from the goal progressively lower. The DAMN arbiter combines each behavior's vote, possibly weighted by a higher level supervisory system. A final smoothing and interpolation step are performed to reduce control switching artifacts caused by discretization within the system. An example of a DAMN system integrating two behaviors' votes is shown in Figure 35.

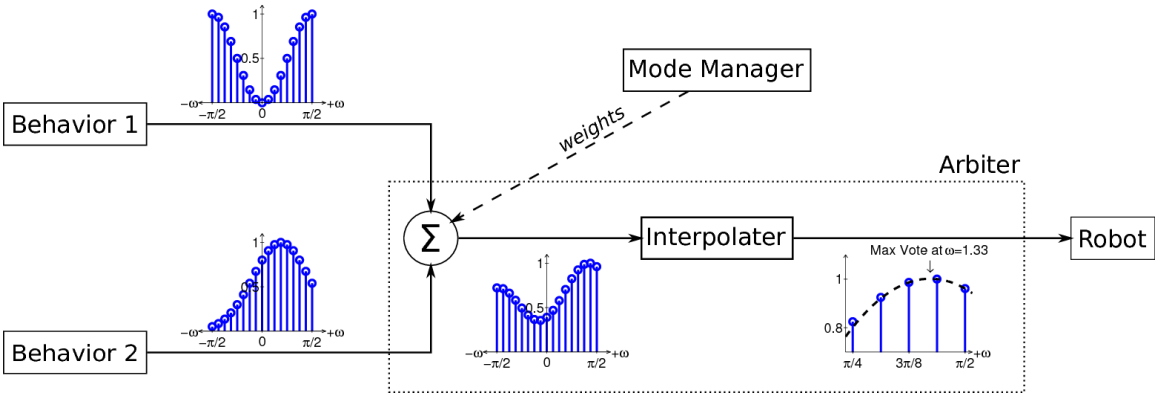


Figure 35: An illustration of the DAMN control scheme integrating two competing behaviors into a single control output.

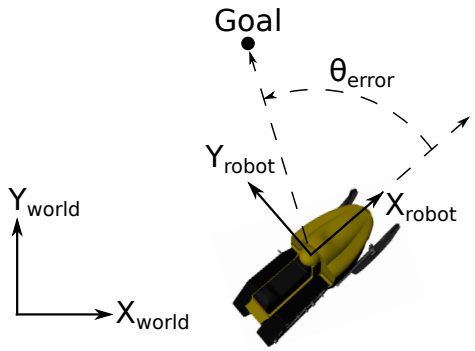


Figure 36: An illustration of the error angle used in the calculation of the angular velocity vote from the goal pursuit behavior module.

3.3.1 Goal Pursuit Behavior

A goal location approximately $100m$ from the rover's position will be provided during simulation trials. A simple goal pursuit behavior has been implemented to convey this information to the DAMN control scheme. An error angle is calculated as the angle between the forward direction in the robot coordinate frame and the vector between the robot and the goal in the world coordinate frame. The calculation of the error angle is illustrated in Figure 36. An experimentally determined gain factor is then applied to convert the error angle into a desired rotation velocity. This step is similar to a simple proportional feedback controller.

To construct the DAMN behavior vote, a Dirac delta function located at the desired rotation velocity is convolved with a Gaussian kernel of unity height. This

places the highest weighted vote on the desired rotational velocity, with neighboring control values having progressively smaller weights. The falloff rate of the weights is controlled by the variance of the Gaussian kernel. A value of $\sigma^2 = 1.5$ has been used in all trials. Equation (26) shows the functional implementation of the weight calculation.

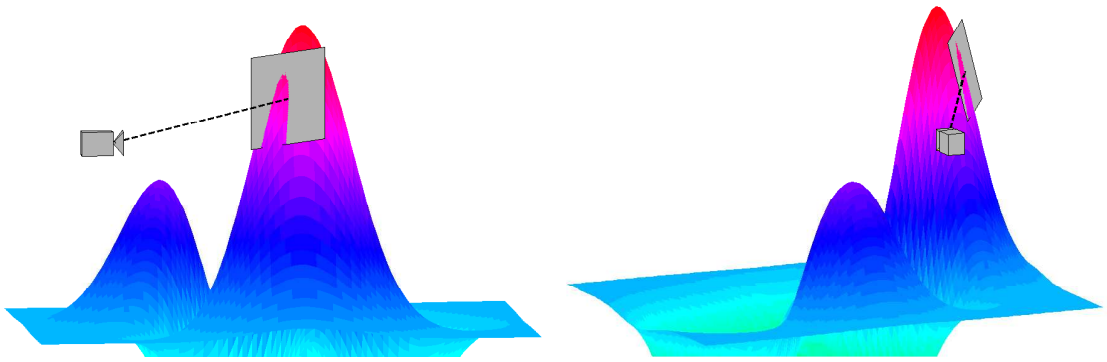
$$weight(\omega) = \delta(\omega - \gamma \cdot \theta_{error}) * e^{-\frac{\omega^2}{\sigma^2}} \tag{26}$$

where γ is the control gain, θ_{error} is the error angle, δ is the Dirac delta function, and σ is the standard deviation of the Gaussian kernel.

3.3.2 Slope Avoidance Behavior

Due to the three dimensional nature of terrain slopes, devising a control behavior that minimizes the robot's chassis roll is not as straightforward as a goal pursuit behavior. n discrete angular velocity controls are selected at regular intervals that span the range of angular velocities supported by the robotic platform. Each of the n potential control laws are simulated forward in time, assuming the rover is on a planar surface and no wheel slip occurs. Since all of the slope information is in image-space, the calculated trajectories are projected into the image. In order to treat each possible trajectory equally, all trajectories are simulated for the same amount of time. As the trajectories for the extreme right and extreme left angular velocities will tend to reach the image edge at an earlier time than the other trajectories, these times are used as the final time for each trajectory. The actual distance each pixel spans on a given trajectory is calculated and stored, as is the yaw angle of the rover at the center of each trajectory pixel. At this point none of the calculations are dependent on the actual environment, and only need to be recalculated if the number of trajectories, n, is changed.

As dense slope estimates are received by the slope avoidance behavior, the slopes



(a) (b)

Figure 37: An illustration of a locally planar patch used to predict the robot's orientation if located at the test pixel. The local plane is aligned with both the terrain slope and the camera ray.

associated with each trajectory pixel, along with the pre-computed trajectory information, are used to calculate the average rover chassis roll over each trajectory. To estimate the chassis roll of a given trajectory pixel, the terrain at a given pixel is assumed to be a plane rotated by the slope angle about an axis extending from the camera focal point to the terrain patch. Figure 37 illustrates this method for a single pixel. The robot normal, Z_{robot} in Figure 36, is aligned with the normal vector of the local terrain plane. By incorporating the projected robot yaw angle with the assumed terrain orientation, an estimate of the full 3-DOF robot orientation is obtained.

To convert the average roll of each trajectory into a DAMN behavior vote, each average roll value is normalized by a predefined maximum acceptable roll value, determined by the capabilities of the robotic platform. The actual weight for each angular velocity is given by Equation (27).

$$weight(\omega) = 1 - \frac{roll_{avg}(\omega)}{roll_{max}} \tag{27}$$

where $roll_{avg}(\omega)$ is the average roll experienced with angular velocity ω, and $roll_{max}$ is the maximum acceptable roll value, determined by the capabilities of the robotic platform.

3.3.3 Simulation Results

To study the effectiveness of the slope information in constructing navigational control laws, the simulation system described in Chapter 2.3 is utilized. Several initial and goal positions were randomly selected across the environment. Traversal distances of at least $100m$ were required to reach the goal in each case. As the focus of these trials is the slope estimation system, ground truth rover poses were extracted from the simulation system and used in place of an on-board localization system. To compare the performance of the slope estimates, each trial has been performed three times. During the first run, no slope information is used and the rover is allowed to drive straight towards the goal. This provides a characterization of the terrain between the start and goal positions. During the second run, ground truth terrain slopes were extracted from the simulation system and provided to the slope avoidance behavior. This demonstrates the best performance possible of the navigation scheme, as the terrain slopes are known exactly. During the third run, the visual slope estimates described in Section 3.2 were used.

Table 3 lists the average chassis roll experienced by the robot during each trial. Using slope information inside the control scheme allowed the average chassis roll to be significantly reduced. Reductions in average roll range from almost 20% in Trial 3, to over 70% in Trial 1. Further, the performance of the slope-aware systems remain largely constant over all trials, with average roll values between 2° and 4°, regardless of the average slope directly between the start and goal positions. The aggregate performance achieved with the slope estimates is similar in all cases to that achieved with the ground truth slopes, further indicating the viability of the estimation system.

Figure 38 illustrates the paths taken by each of the three systems during Trial 1, superimposed on a terrain contour map. As is evident in this figure, the rover trajectories when incorporating slope information tend to be nearly perpendicular to the terrain contour lines. In this configuration, the rover chassis roll is minimized.

Table 3: Average Chassis Roll During Simulation Trials

Trial Number	Chassis Roll Driving Directly To The Goal	Chassis Roll Using Ground Truth Slopes	Chassis Roll Using Visual Slope Estimates
1	11.07°	2.43°	2.55°
2	6.05°	3.58°	3.64°
3	3.99°	3.25°	3.30°
4	11.81°	5.32°	6.04°
5	8.93°	4.18°	4.51°

However, it is impossible to reach the goal and simultaneously maintain an optimal roll orientation. Consequently, the rover periodically changes direction, experiencing an increase in chassis roll only briefly. This is similar to performing "switch-backs" when driving up a mountain side. It should be noted that no penalty for total distance traveled has been imposed in this system.

3.3.4 Conclusions

In glacial environments, one of the major hazards for autonomous robots is the terrain slope. In Section 3.2, a method for visually estimating the terrain slope in front of the rover is presented. However, these estimates only provide the slope in a single plane, instead of a full 3D slope characterization. Despite this limitation, a simple reactive controller has been implemented that makes use of the 2D slope estimates to avoid terrain that would induce large amounts of chassis roll. This controller has been tested within the simulation system described in Chapter 2.3, and shown to be visually similar to real glacial environments in Section 3.2. During these tests, the control scheme was able to significantly reduce the average chassis experienced during a traverse to a goal location, by over 70% in some cases. Additionally, it has been demonstrated that the paths resulting from use of the visual slope estimates are similar to those resulting from ground truth slope information, indicating the viability

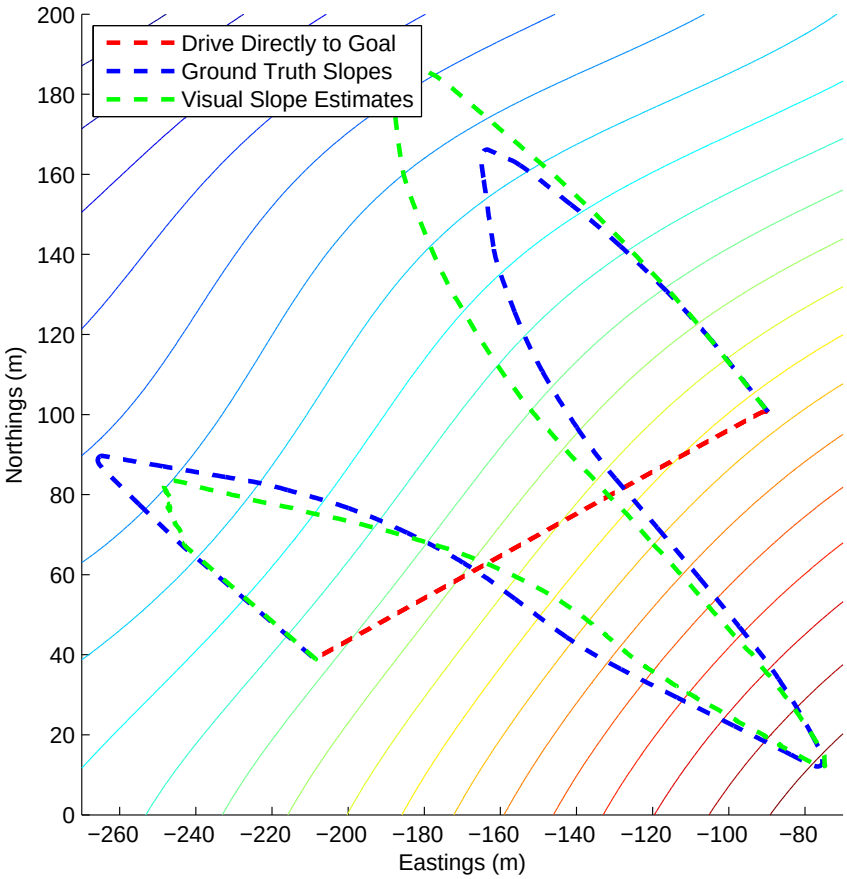


Figure 38: An illustrates the differences in the rover paths based on using no slope information (red), ground truth slope information (blue), and visual slope estimates (green).

of the slope estimate technique as a valuable input to a glacial hazard avoidance system. However, this system merely demonstrates one possible control scheme, in which a single aspect of the rover's situation is considered. Further development of the control scheme in future is expected to include more aspects of the rover's state in the decision process as well as incorporate a path planner that is capable of maintaining a map of slope data over time.

CHAPTER IV

VISION-AUGMENTED LOCALIZATION

When high-accuracy localization is needed, robotic systems typically rely on augmented GPS systems, such as differential GPS (D-GPS) or real-time kinematic GPS (RTK-GPS), which are capable of centimeter-range accuracy. If the system also requires orientation information, high quality inertial measurement units (IMUs) are integrated into the localization subsystem. For example, the autonomous driving agents developed for competition in the various DARPA Grand Challenge races include localization systems based on the fusion of many high-accuracy sensors [67, 113, 115]. However, popular high-accuracy sensing equipment, such as the Applanix GPS and IMU, typically costs in excess of $50k [26]. For a multi-agent robotic network, such sensing equipment is prohibitively expensive. Instead, the localization subsystem must make due with consumer-grade GPS receivers, which have a typical accuracy of $10m$. If a more precise positional fix is needed for mapping or satellite validation applications, or if the robot orientation is desired, additional sensor data must be incorporated.

One possible source of additional localization data is the robot's vision system. Recent work in the area of vision-based odometry [89], and visual SLAM (simultaneous localization and mapping) [24] have shown even single-camera vision systems capable of generating reliable localization estimates. At their core, these systems match distinctive image features between sequential video frames, and use the camera geometry to estimate the camera's motion. Consequently, one of the prerequisites for vision-based localization is the ability to extract and match image features between frames. This is generally done by using one of a number of keypoint detectors,

such as Harris [49] or SIFT [73]. These detectors rely on finding pixels with strong directional gradients. In an office environment, sharp corners in furniture and shelving provide ideal candidates, while the irregular edges in rocks and outcropping in desert environments also serve as viable choices. However, glacial environments generally lack these types of distinctive features (see Chapter 1.1 for example images). To that end, a set of image preprocessing steps have been developed to boost the feature extraction performance, and a study of the detection performance of common feature detection algorithms has been conducted. Section 4.2 describes the results of naively applying standard feature detectors to glacial images, presents the developed preprocessing procedure that dramatically improves both the feature extraction and feature matching performance in this domain, and compares the performance of five common features detectors on the preprocessed image set. Final feature detector selection is based both on extraction performance and execution time.

Despite the performance gains over standard wheel odometry [89], visual odometry and visual SLAM systems are still incremental; the current position estimate is updated based on a change in observations. One of the fundamental issues when using any incremental system is localization drift. As the system runs, small errors accumulate, resulting in significant localization error over time. If the robot path contains loops, a procedure known as loop closure can update the localization information of past positions, significantly reducing the total experienced drift. Even with these additions, as the traversed path becomes longer, the localization performance degrades. Further, as a camera converts three dimensional world points into two dimensional image pixels, each pixel represents a direction only. The general solution to localization using bearing-only observations is to incorporate metric information from an external source. In order to build a drift-free localization subsystem, the low-accuracy GPS data has been fused with the incremental localization of vision-based SLAM. Section 4.1 introduces the basic probabilistic framework used in SLAM

systems, as well as discusses some of the modifications necessary to implement SLAM with bearing-only measurements provided by vision sensors. Section 4.3 presents the details of the GPS-fused system, based on a FastSLAM [81] infrastructure. The results of this system applied to both simulated and real terrain are then presented in Section 4.4.

4.1 Visual Odometry Techniques

The use of vision for localization has gained traction recently, and revolves mainly around the use of multi-view geometry techniques and the related simultaneous localization and mapping (SLAM) methods. To employ multi-view geometry using a single camera, multiple images are acquired at different points in time. Image features in one image are then matched to features in the other images. With a sufficient number of point correspondences, the geometric transformation between the two images may be extracted. In contrast, vision-based SLAM systems seek to estimate the 3D position of extracted image features. These position estimates are then projected into the camera image space of subsequent frames, and matched with image features in the new frame. The image space error between the projected landmark and the new image feature is used to update the position estimate of the landmark. Implicit in both systems is the ability to reliably extract and match image features between frames.

Once distinctive features have been extracted and matched between frames, the point correspondences are sent to a visual localization system. The most prominent use of vision for localization revolves around multi-view geometry methods and the related simultaneous localization and mapping (SLAM) methods. To employ multi-view geometry for localization [51,89], multiple images are captured at different points in time. Assuming the rover is actively moving during this period, the images will be of the same scene, but from slightly different viewpoints. If features can be extracted

and matched between the images, then a triangulation system can be constructed using the visual features as landmarks instead of stars or mountains. The main difference in this system is that the landmark positions are also unknown.

If we assume that the intrinsic camera parameters (focal length, principle point, etc.), K, are known and fixed, then all point correspondences in the image can be converted into normalized image coordinates, $\hat{x}$, by pre-multiplying the inverse camera matrix, K, as in Equation (28).

$$\hat{x} = K^{-1} \cdot x \tag{28}$$

All normalized point correspondences must satisfy the epipolar constraint in Equation (29), where E is known as the essential matrix. The essential matrix encodes the rigid body transformation (rotation and translation) of the camera from the initial pose to the new pose.

$$\hat{x}_2^T E \hat{x}_1 = 0 \tag{29}$$

$$E = [t]_\times R \tag{30}$$

where $[t]_\times$ is the skew symmetric matrix that represents a vector cross-product with the translation vector, t, and R is the rotation matrix.

This constraint is phrased in terms of the image plane quantities, $\hat{x}_1$ and $\hat{x}_2$. When the 3D point, X, is projected into the two image planes, then the image point, $\hat{x}_2$, must lie on the ray $E\hat{x}_1$ in image 2 (the distance or dot product is zero). In essence this means that since the 3D point, X, creates a unique set of point correspondences in the two image planes, the actual 3D location of X is not needed to recover the rigid body transformation. If E is known, the rotation and translation, up to scale, may be extracted using methods such as Singular Value Decomposition (SVD) [51]. With careful calculation, all results can be represented in the same scale, although

the exact metric value of this scale is still unknown. External information, such as wheel odometry, may be applied to estimate the true metric scale value.

In contrast, vision-based SLAM attempts to solve the same problem, but does so in a probabilistic framework [6, 32]. The system attempts to maximize the joint probability of the robot pose, x_t, and the map of 3D landmarks, m, given the entire set of robot control inputs, $\mathbf{u}_{0:t}$, and observations, $\mathbf{z}_{0:t}$, shown in Equation (31).

$$\arg\max_{x_t,m} \left(p(x_t, m|\mathbf{z}_{0:t}, \mathbf{u}_{0:t})\right) \tag{31}$$

Two major advantages arise from this approach. First, since the landmark positions are estimated in 3D world space, the 2D projections can be calculated directly, intrinsically satisfying the epipolar geometric constraints. Secondly, as the solution to the SLAM problem is a probability distribution, an estimate of the error of either the robot pose or the landmark positions may be obtained by marginalizing out the appropriate variables. These error estimates can then be projected into the camera frame, limiting the size of the search region for corresponding feature points. In contrast, propagating an error estimate through the 10th order root solving method required in the multi-view geometry approach is simply not practical.

Unfortunately, Equation (31) does not suggest a solution; additional assumptions must be made. Generally, two mathematical models are introduced, one that propagates the robot pose forward in time, and one that predicts observation values based on the current information. These are shown in Equations (32) and (33) respectively.

$$p(x_t, m|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t}) = \int p(x_t|x_{t-1}, u_t) \cdot p(x_{t-1}, m|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t-1}) \cdot dx_{t-1} \tag{32}$$

$$p(x_t, m|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}) \propto p(z_t|x_t, m) \cdot p(x_t, m|\mathbf{z}_{0:t-1}, \mathbf{u}_{0:t}) \tag{33}$$

With this factorization, the robot pose distribution is predicted at time t by Equation (32), incorporating the new control input, u_t. This prediction is then corrected

using the information from the new observations, z_t, by Equation (33). This recursion is easily implemented as an extended Kalman filter (EKF), where the EKF state is a concatenation of the robot pose, x, and the landmarks, m. However, by augmenting the EKF state vector with landmark positions, the complexity of the system grows as $O(n^2)$ in the number of landmarks. This is due to EKF keeping a full covariance matrix of the state. This is only feasible for systems that need a small number of landmarks. For example, systems that only operate within a single room will see the same set of landmarks repeatedly, and, with careful selection, a relatively small set can cover the entire workspace. The most notable implementation of this approach is the MonoSLAM algorithm [24].

A commonly used alternative is to employ a Rao-Blackwellized particle filter (PF) to estimate the robot pose [81], also known as FastSLAM. The PF samples many pose "particles" from the pose distribution, and assumes each pose particle is the true robot pose. Since the error in the robot pose is now assumed to be zero, the landmark distribution estimates become decoupled, as shown in the factorization in Equation (34).

$$p(x_t, m|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}) = p(x_t|\mathbf{z}_{0:t}, \mathbf{u}_{0:t}) \cdot \prod_{i=1}^{M} p(m_i|\mathbf{x}_{0:t}, \mathbf{z}_{0:t}) \tag{34}$$

where M is the current number of landmarks in map, m.

Similar to the EKF-SLAM system, the landmark states are estimated using an EKF. Only, in this formulation, adding a landmark consists of adding an independent, low-dimension EKF to the system, instead of augmenting a single, large EKF. As each low-dimension EKF may be seen as a constant time operation, the FastSLAM system time complexity is only $O(n)$ in the number of landmarks. However, implicit in this method is the assumption that the space of all possible camera trajectories can be properly sampled, leading to issues of sample impoverishment. Certain refinements can be made to reduce these issues, but successful loop closures in FastSLAM remain

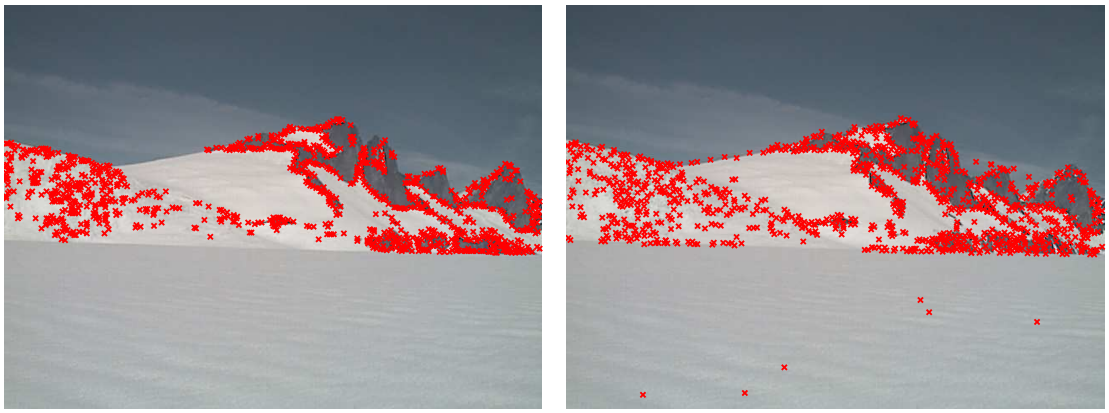
an issue. Also, the FastSLAM system was not originally derived for use with vision, though vision-based implementations have recently been presented [33, 93].

4.2 Glacial Image Feature Extraction

As discussed in the previous section, the ability to extract and match distinctive image features between frames is crucial for the proper operation of a vision-based localization system. One of a number of common keypoint detectors are generally used to meet the data needs of real-time visual odometry systems, such as Harris [49] or SIFT [73]. These detectors rely on finding pixels with strong directional gradients. In an office environment, sharp corners in furniture and shelving provide ideal candidates, while the irregular edges in rocks and outcropping in desert environments also serve as viable choices. However, glacial environments generally lack these types of distinctive features. As seen in the sample glacial photograph in Figure 39, if we focus only on the traversable foreground section, the image consists of predominantly white, dune-like snow structures. Some subtle color variations are visible, caused by the surface texture and shadows, but large color gradients in the foreground are not present [131]. For illustrative purposes, the Harris detector and SIFT algorithm have been applied to this sample image acquired from field tests on Mendenhall Glacier in Alaska. While the mountains in the background produce an ample number of features from both methods, almost no features have been detected in the snowy foreground.

4.2.1 Preprocessing

Since standard feature detectors search for pixels exhibiting strong directional gradients, the foreground image gradient must be boosted for these detectors to perform properly. Ideally, the image enhancement should be non-uniform, adaptively enhancing the foreground regions while leaving areas of sufficient contrast alone. The contrast-limited adaptive histogram equalization (CLAHE) stage utilized in the slope

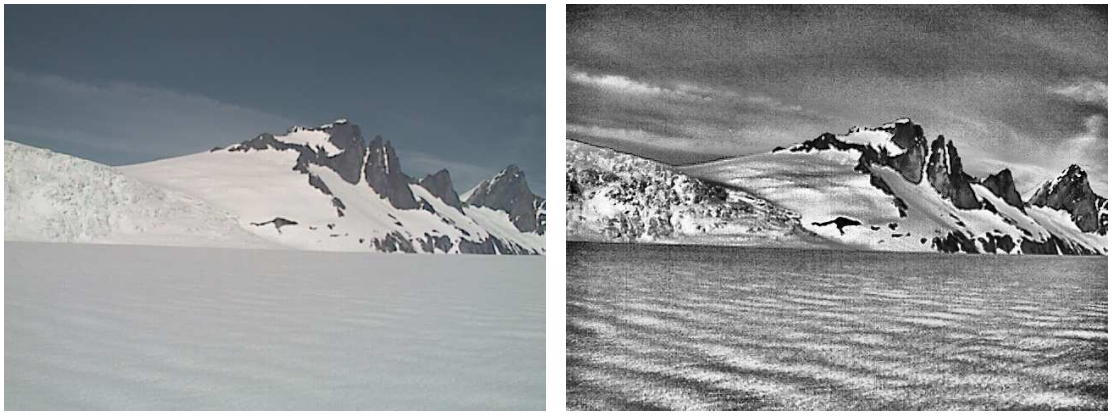


(a) (b)

Figure 39: The results of applying (a) the Harris corner detector and (b) the SIFT algorithm to a sample glacial image from the May/2009 dataset on Mendenhall Glacier. While the mountains in the background produce a significant feature volume, the foreground is largely devoid of features.

estimation system of Chapter 3.2 possesses such properties. Additionally, a local enhancement limit is present, which reduces issues of over-enhancement of noise. The results of CLAHE enhancement of the sample glacial image is shown in Figure 40. As the CLAHE-enhanced image is computed as part of the slope estimation system, it may be used within the feature extraction subsystem at no computational cost.

To test the performance of this preprocessing step, a set of sequential images were selected from five of the test sites on Mendenhall Glacier. SIFT features and descriptors were extracted for each image, one set from the original image and a second set from the enhanced image. The feature descriptors are used as input to a least-euclidean-distance matching algorithm to find correspondences. As a check for matching consistency, a least-squares method [88] was used to estimate the essential matrix, E, which describes the motion of the camera between consecutive frames. The set of inliers from the best essential matrix estimate are considered consistent matches. An example of point matches between two consecutive frames is shown in Figure 41. Extracted features are marked on each image, correct feature matches are drawn between features in green, while incorrect matches are drawn in red.



(a) (b)

Figure 40: (a) A sample glacial image from the May/2009 dataset on Mendenhall Glacier, and (b) the results of applying CLAHE adaptively enhances the foreground regions while leaving areas of sufficient contrast alone.

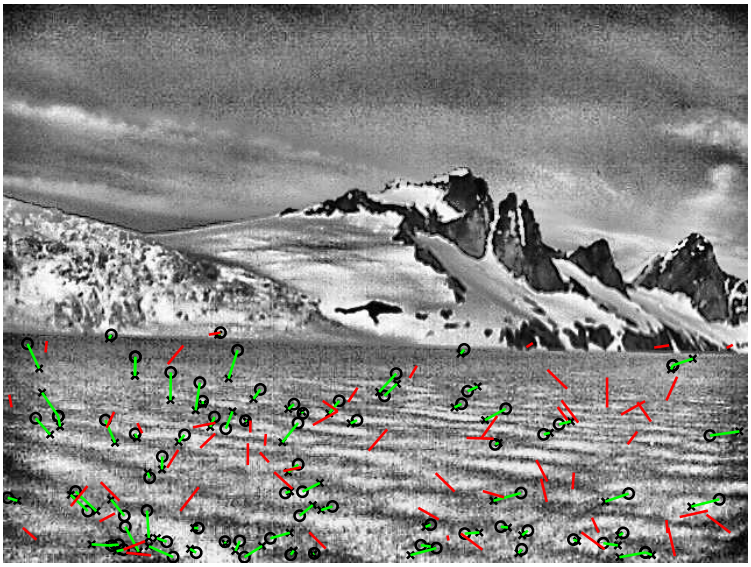


Figure 41: An example of feature extraction and matching between two consecutive frames. Extracted features from frame 1 and frame 2 are marked on frame 1 with an 'x' and 'o' respectively. Correct feature matches, as determined by estimating the essential matrix, are drawn between features in green, while incorrect matches are drawn in red.

Table 4: Image Preprocessing Performance Based On SIFT Feature Matching, Averaged Per-Frame Results

Site		Feature Count	Match Count	Inlier Count
A	raw	75.5	37.2	10.3
	enhanced	4736.9	1972.8	396.1
B	raw	362.2	119.2	18.5
	enhanced	2410.5	850.6	129.5
C	raw	509.2	399.0	303.9
	enhanced	4007.6	2497.6	1357.2
D	raw	79.2	39.6	16.9
	enhanced	3243.5	1330.9	257.0
E	raw	128.9	59.4	28.2
	enhanced	3445.6	1664.8	575.4

As the central regions of glacial ice sheets are devoid of mountain peaks and other structures (see Chapter 1.1 for details), it is desirable to test the system performance utilizing only the foreground features. To facilitate this, the background structures have been masked out using the horizon line detection scheme presented in Chapter 3.1, and only features within the masked region are considered.

Averages for the raw and enhanced images of each data set have been generated concerning the number of feature points extracted, the number of feature points matched, and the number of correctly matched features as determined by the essential matrix estimate. For the purposes of this comparison, the SIFT detector has been used with the default parameters. The results are presented in Table 4.

In order for visual odometry methods to operate correctly, many visual features are required that cover the whole image area. If all of the points used to calculate the essential matrix, E, are drawn from the same image region, the algorithm degenerates, resulting in a near-singular condition. Although SLAM systems have no

absolute minimum number of required features, as the number of observations decreases, the resulting state estimate reverts to a dead-reckoning system. In such a scenario, divergence from the true state values is likely. For a robust estimate of the robot state, it is important to have a reasonably large pool of extracted features.

When extracting features from the unprocessed images, a relatively small number were captured from the foreground region at most sites. After the matching process and outlier rejection, only Site C consistently produced more than 30 correct matches. Site C was located on the glacier's terminus. Here, the terrain consisted largely of exposed blue ice, which provides far greater texture and color variation than the snow covered areas of the other sites. Consequently, the feature detection phase performs well on the raw images, though the image enhancement still provides substantial improvement. The other regions were located towards the center of the different glacier branches, or in the large expanse before the glacier divides into branches in the case of Site A. These areas are marked by slowly varying elevations and unbroken surface snow, which produce few high-quality features. By employing the proposed enhancement procedure, the density of features increased by a factor of five to ten, with the number of between-frame matches improved by a similar amount. In all cases, the enhanced images produced over 100 feature matches, which should be sufficient to ensure feature coverage over viewable foreground.

4.2.2 Feature Detector Selection

Feature detectors are often evaluated based on a set of invariance properties. Rotational invariance implies that a feature will be detected regardless of the camera's roll. Harris features use an eigenvalue decomposition to ensure rotational invariance [49]. Rotation invariance is achieved in a similar fashion in both the Scale-Invariant Feature Transform (SIFT) [73] and the Speeded Up Robust Features (SURF) [8] algorithms. The Maximally Stable Extremal Regions (MSER) detector [76] searches for image

regions with well-defined boundaries instead of single pixel features. By utilizing the shape of the region, it achieves full affine transformation invariance, which includes simple rotations. Finally, scale invariance means that a feature will be detected regardless of underlying landmark's distance from the camera. SIFT and SURF achieve scale invariance by searching for features in downsampled versions of the source image. Since an MSER feature can be any size within the image, a simple size normalization produces a scale-invariant feature. An exhaustive treatment of the level of invariance of the different feature detectors in real images may be found in [78].

However, the importance of the various invariance properties is dependent on the application. For object recognition from a prototype image, affine-invariance is important as the object's orientation in the environment will not be known. However, for matching sequential images acquired from a mobile robot, affine-invariance is of little importance, as the view change between frames will be minimal. In fact, rotational invariance may not even be required as the robot will be basically upright during operation. Scale invariance, on the other hand, is important. As the robot drives forward, the visual landmarks will get closer to the robot, increasing their size in the image. A scale-invariant feature would be able to find and match these landmarks over the traverse. Additionally, larger scale surface features, such as the alternating dark and light linear streaks on the sample image in Figure 40b can act as features if extracted at an appropriate scale.

While the theoretical analysis indicates that either SIFT or SURF features are good candidates for a ground-based visual localization system, the ultimate detector selection should be based on field test results. The four common feature detectors described previously have been selected for comparison. These detectors span the gamut of current keypoint extraction techniques. The Harris corner detector is the least complex detector tested, resulting in the shortest execution time. SIFT is perhaps the most widely used feature detector, which has the advantageous property of

scale-invariance. The SURF feature detector purports similar extraction performance to the SIFT algorithm while reducing the computation time. Finally, the MSER detector has been tested, which searches for well-defined regions as opposed to single points.

Each feature detector has been applied to the enhanced version of the sequential images from the five different test sites on Mendenhall Glacier. The SIFT descriptor was generated for each extracted point to aid in appearance-based matching. These descriptors were used as input to a least-euclidean-distance matching algorithm. As before, the matched points have been used to estimate the essential matrix. Any matching pair that deviates significantly from the essential matrix estimate is marked as an outlier. Averages for each algorithm and data set concerning the number of feature points extracted, the number of feature points matched, and the number of correctly matched features as determined by the essential matrix are presented in Table 5.

From Table 5 it is clear that the SIFT and SURF detectors outperform both the Harris and MSER detectors across all test sites. The lack of scale-space extraction for the Harris operator hinders its performance as few of the SIFT features extracted were at a scale of 1.0. The MSER detector is simply a poor fit for this domain of images. While the alternating texture components of the enhanced image seem likely feature candidates, the smooth edge gradient of these regions do not produce a stable region size over different intensity thresholds, and are eliminated by the MSER algorithm. Although the SURF detector generally produced significantly fewer single-frame features than the SIFT algorithm, the matching performance was generally similar to SIFT. This gives SURF a computation advantage in two ways. First, the SURF detector requires approximately half the computation time of the SIFT algorithm on a per-feature basis. Second, the total feature computation time and feature matching time are proportional to the number of extracted features. By

Table 5: Feature Detector Performance, Averaged Per-Frame Results

Site		Feature Count	Match Count	Inlier Count
A	harris	888.0	306.2	46.6
	mser	466.2	171.4	24.0
	sift	4736.9	1972.8	396.1
	surf	1995.3	1063.8	281.6
B	harris	1000.0	308.9	50.1
	mser	293.1	90.9	11.2
	sift	2410.5	850.6	129.5
	surf	2111.6	816.4	147.6
C	harris	350.3	214.7	126.9
	mser	349.4	189.5	68.4
	sift	4007.6	2497.6	1357.2
	surf	2014.5	1541.5	1053.1
D	harris	216.6	83.5	12.0
	mser	390.8	139.5	17.3
	sift	3243.5	1330.9	257.0
	surf	1767.0	831.3	237.7
E	harris	297.2	139.5	31.8
	mser	401.4	171.6	37.3
	sift	3445.6	1664.8	575.4
	surf	2012.0	1117.0	480.0

reducing the per-frame feature count, a significant speed improvement is possible without sacrificing matching performance. This makes SURF the clear choice for this application.

4.3 Localization System Implementation

Despite the seemingly simple mathematical description of SLAM presented Equation (31), there are many implementation issues that must be resolved before the system will operate successfully. For example, the map, expressed solely as m in the equation, requires mechanisms for representing, adding, removing, and searching landmarks. Details of various aspects of the implemented SLAM system, including the sensor fusion method for incorporating the low-accuracy GPS data, are described in the following.

4.3.1 Filter Selection

The advantages of EKF SLAM or PF SLAM over the other is specific to the situation. In this application, each rover is expected to navigate from a known location (base camp), to its desired goal position defined by the scientist. The distance from base camp to goal position is likely to be several kilometers. Under these conditions, a large number of landmarks will be required to localize the rover over very long traverses, even if only a small subset of landmarks are visible at any instance. The PF SLAM implementation allows a large amount of flexibility in this situation. Large databases of landmarks are possible using the PF approach, with one lidar implementation able to operate in real-time with a database of 50,000 landmarks [82]. Different sorting and indexing strategies can be implemented to tailor the database system to the application. In contrast, the EKF SLAM approach maintains all the landmarks in a single covariance matrix. This limits the number of landmarks to the order of 100 if real-time performance is required [24].

Further, the expected path of the rover from start to goal is likely to be straight,

or at least piecewise straight under the presence of obstacles or untraversable terrain. Figure 38 in Chapter 3.3 shows an example of a piecewise straight path implemented in response to ground slope conditions. Few, if any, loops will be executed during these traverses, mitigating the issues related to loop closures in PF SLAM systems. For these reasons, the PF SLAM approach has been selected for the localization subsystem of the SnoMote project.

Since each particle used to approximate the robot state distribution will interpret the image with respect to its assumed state, each pose particle will create different landmarks and make different data associations. Consequently, at time t, the n^{th} particle contains a current robot state, x_t^n, its own map, $\boldsymbol{m}^n$, and a particle weight, w^n. In this context, a map consists of a series of i landmark position estimates, μ_t^i, and corresponding error covariances, Σ_t^i. When a new image is captured from the camera system, the feature extraction subsystem described in Section 4.2 produces a set of J observations in image-space. This only needs to be done once, not on a per-particle basis. Then, for each particle, a new robot state is sampled from the proposal distribution, conditioned on the particle's previous state, x_{t-1}^n, and the current robot control commands, u_t. Within Rao-Blackwellized particle filters, this pose is assumed to be correct for that particle. Using the new robot state, the image-space coordinates of each landmark within the map, $\hat{z}_t^i$, are obtained using the measurement function, $h(\cdot)$, and a linear uncertainty estimate in image-space, Z_t^i, is obtained using the Jacobian of the measurement function with respect to the landmark. It should be noted that the choice of landmark representation does not affect the sequence of steps, but rather the specifics of the measurement function, $h(\cdot)$. Data association is then performed, matching extracted image features with the predicted landmark positions. Each of the i landmarks are then updated using the standard EKF equations [4], and the probability of that association, p_t^i, is calculated. For features converted into new landmarks and landmarks that were unobserved, a fixed probability estimate is

used. Finally, the weight for the current particle is calculated as the product of the individual data association probabilities, and the particles are resampled according to the Sequential Importance Resampling (SIR) algorithm [70]. The basic steps for a PF SLAM iteration are shown in Algorithm 1.

4.3.2 Robot Parametrization

As the robot is moving on a three dimensional surface, the state of the robot consists of at least a three dimensional position, $\vec{x} = (x, y, z)^T$, and a three dimensional orientation, q. A quaternion orientation representation has been selected, as it provides a more numerically stable method of handling incremental rotations than a roll-pitch-yaw representation [2], and requires fewer state variables than a full rotation matrix representation. A quaternion that expresses a pure rotation must be unit length, but numerical errors in the rotation propagation can cause the quaternion length to drift. For these reasons, the quaternion value is normalized after each update. The final state vector is $x_t^n = (\vec{x}_t, q_t)^T$. The control commands, u_t, provided to the robot are in the form of a linear velocity, v, and a yaw angular velocity, ω.

During each update step, a state particle samples a new state from the proposal distribution, $p(x_t^n | x_{t-1}^n, u_t)$. The proposal distribution is approximated by a multivariate Gaussian distribution, as shown in Equation (35). The values σ_x^2, σ_y^2, and σ_z^2 are the uncertainty variances for the robot position, potentially caused by track slippage and uneven terrain. The uncertainty variance for the robot's orientation are represented by σ_{q1}^2, σ_{q2}^2, σ_{q3}^2, and σ_{q4}^2. This accounts for changes in roll and pitch induced by the uneven terrain. Uncertainties in the execution of the commanded linear and angular velocities caused by slippage in the track and steering linkage are accounted for in σ_v^2 and σ_ω^2.

Algorithm 1 Particle Filter SLAM Update Loop [79]

Require: $\boldsymbol{X}_{t-1} = \left\{ \left(x_{t-1}^n, \boldsymbol{m}^n, w^n \right) \mid \forall n \in 1 \dots N \right\}$ * The set of particles at t-1
Require: $\boldsymbol{m}^n = \left\{ \left(\mu_{t-1}^i, \Sigma_{t-1}^i \right) \mid i = 1 \dots I \right\}$ * The map at time t-1
$\left(z_t^1, \cdots, z_t^J \right) = ExtractFeatures(image)$ * Extract features from the new image
for $n = 1$ **to** N **do** * Loop over particles
$x_t^n \sim p(x_t | x_{t-1}^n, u_t)$ * Sample the new robot pose
$\hat{z}_t^i = h\left(x_t^n, \mu_{t-1}^i \right) \forall i \in 1 \dots I$ * Predict the landmark measurements
$H_{m,t}^i = \frac{\partial}{\partial m} h\left(x_t^n, m \right)|_{m=\mu_{t-1}^i} \forall i \in 1 \dots I$
$Z_t^i = H_{m,t}^i \Sigma_{t-1}^i {H_{m,t}^i}^T + R_t$
$DA = DataAssociation(\hat{z}_t^{1 \dots I}, z_t^{1 \dots J})$ * Associate predictions with observations
for $\left(\hat{z}_t^i, z_t^j \right) \in DA$ **do**
$K = \Sigma_{t-1}^i {H_{m,t}^i}^T Z_t^{i-1}$ * Update landmark EKF
$\mu_t^i = \mu_{t-1}^i + K(z_t^j - \hat{z}_t^i)$
$\Sigma_t^i = \left(I - K H_{m,t}^i \right) \Sigma_{i,t-1}$
$p_t^i = |2\pi Z_t^i|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (z_t^j - \hat{z}_t^i)^T Z_t^{i-1} (z_t^j - \hat{z}_t^i) \right)$ * The association probability
end for
for $\hat{z}_t^i \notin DA$ **do**
$\mu_t^i = \mu_{t-1}^i$ * Unobserved landmarks remain unchanged
$\Sigma_t^i = \Sigma_{t-1}^i$
$p_t^i = p_{missing}$ * Use a fixed 'missing' probability
end for
$k = 0$
for $z_t^j \notin DA$ **do**
$k = k + 1$
$\mu_t^{I+k} = h^{-1}(x_t, z_t^j)$ * Create landmarks from new observations
$H_{m,t}^{I+k} = \frac{\partial}{\partial m} h\left(\hat{x}_t, m \right)|_{m=\mu_t^{I+k}}$
$\Sigma_{I+k,t} = \left({H_{m,t}^i}^T R^{-1} H_{m,t}^i \right)^{-1}$
$m = m \cup \left(\mu_{I+k,t}, \Sigma_{I+k,t} \right)$ * Insert new landmark into the map
$p_t^{I+k} = p_{new}$ * Use a fixed 'insertion' probability
end for
$w^n = \prod_{1 \le i \le I} p_t^i \cdot \prod_{I+1 \le j \le I+k} p_t^j$ * Calculate particle weight
end for
$\boldsymbol{X}_t = Resample(\boldsymbol{X}_{t-1})$ * Resample particles proportional to the particle weight

$$p(x_t^n | x_{t-1}^n, u_t) \sim \mathcal{N}(\mu_t(x_{t-1}^n, u_t), \Sigma_t(x_{t-1}^n, u_t)) \tag{35}$$

$$\mu_t(x_{t-1}^n, u_t) = \begin{pmatrix} \vec{x}_{t-1} + q_{t-1} \begin{pmatrix} \Delta t \cdot v \\ 0 \\ 0 \end{pmatrix} q_{t-1}^{-1} \\ q_{t-1} \begin{pmatrix} \cos(\Delta t \cdot \omega / 2) \\ 0 \\ 0 \\ \sin(\Delta t \cdot \omega / 2) \end{pmatrix} \end{pmatrix} \tag{36}$$

$$\Sigma_t(x_{t-1}^n, u_t) = \Lambda + J_u \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} J_u^T \tag{37}$$

$$J_u = \frac{\partial}{\partial u_t} \mu_t(x_{t-1}^n, u_t) \tag{38}$$

where $\Lambda = diag(\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_{q1}^2, \sigma_{q2}^2, \sigma_{q3}^2, \sigma_{q4}^2)$.

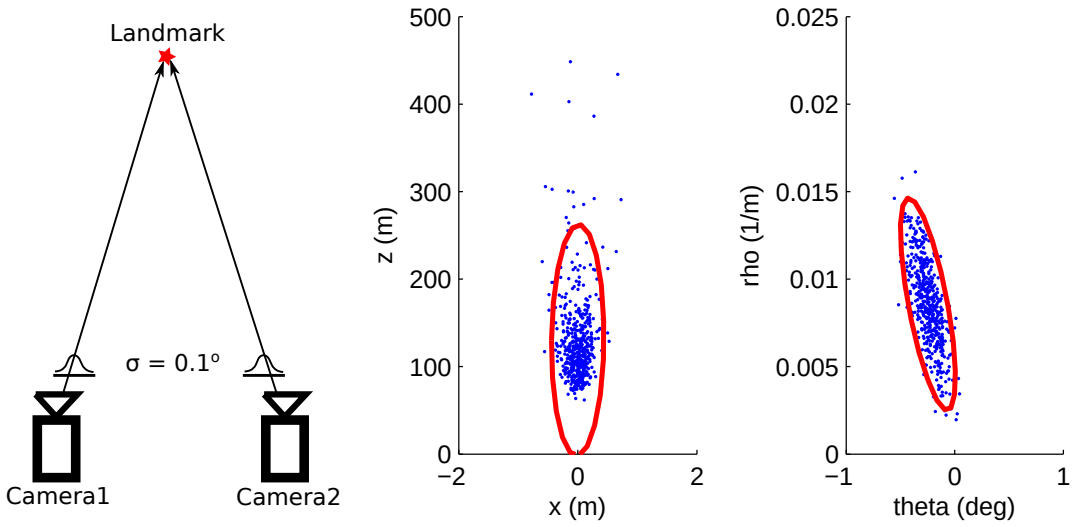
4.3.3 Landmark Parametrization

In a PF SLAM implementation, each landmark is represented as an independent EKF. The measurement function, $h(\cdot)$, predicts the observation that will be generated from a given landmark. For camera-based SLAM, this measurement function implements the pinhole camera model equation, shown in Equation (39) to convert the 3D world coordinates of a landmark into the 2D pixel coordinates of the camera image. The landmark position is then updated using the error between the landmark projection and the current observation in image-space.

$$\begin{pmatrix} u \\ v \end{pmatrix} = h \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} m_x \cdot X \\ m_y \cdot Y \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{39}$$

where $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ is the location of a point in 3D world coordinates, $\begin{pmatrix} u \\ v \end{pmatrix}$ is the location of the same point in image pixel coordinates, f is the camera focal length, m_x and m_y are scale factors that convert pixel indices into metric distances on the image plane, and $\begin{pmatrix} u_0 \\ v_0 \end{pmatrix}$ is the principle point in pixel coordinates, ideally in the center of the image.

During the update phase, the nonlinear measurement model must be linearized around the current landmark position, with the accuracy of the linear approximation dependent on how nonlinear the measurement function is at that point. Using the



(a) (b) (c)

Figure 42: (a) A simple 2D example showing a landmark observed from two different camera locations. The bearing observation of each camera is corrupted with a small amount of Gaussian noise. (b) The estimated landmark position plotted in the X-Z plane with a 2-σ ellipse. (c) The estimated landmark position plotted in the ρ-θ plane with a 2-σ ellipse. [83]

natural landmark parametrization of $(X, Y, X)^T$, the measurement function can exhibit pronounced nonlinearities, particularly when the parallax between consecutive camera frames is small. Using an inverse depth parametrization, the nonlinearities of the measurement function are greatly reduced [83]. A simple 2D example is shown in Figure 42 in which a landmark is observed from two different camera positions with Gaussian noise corrupting the bearing measurement from each camera. The resulting landmark position estimates are then plotted using the (X, Z) parametrization and a (ρ, θ) parametrization. The distribution of landmark positions with the (ρ, θ) parametrization is much better characterized with a Gaussian model than the (X, Z) parametrization distribution.

Additionally, if the inverse depth, ρ, spans a range $[0, 1]$, then the corresponding depth spans the range $[1, \infty)$. This facilitates applying large uncertainty regions to the initial landmark position within the landmark state covariance matrix. However,

the azimuth-elevation-inverse depth parametrization for a landmark can only be interpreted if the original observation position is known. Since there is no uncertainty in the robot state in the PF implementation, the observer position is simply stored as a parameter of each landmark, and not as an EKF state variable. The final landmark state representation is $(\theta, \phi, \rho\,; x, y, z)^T$, where $(x, y, z)^T$ is the camera location at which the landmark was originally observed, and $(\theta, \phi, \rho)^T$ is the elevation, azimuth, and inverse depth of the observation respectively. The measurement function for this parametrization is shown in Equation (40).

$$\begin{bmatrix} u \\ v \end{bmatrix} = h\left(\theta, \phi, \rho\,; x, y, z\right) = \frac{f}{Z} \begin{bmatrix} m_x \cdot X \\ m_y \cdot Y \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{40}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = q_C \begin{bmatrix} \frac{1}{\rho}\cos(\theta)\cos(\phi) + x - x_C \\ \frac{1}{\rho}\cos(\theta)\sin(\phi) + y - y_C \\ \frac{1}{\rho}\sin(\theta) + z - z_C \end{bmatrix} q_C^{-1} \tag{41}$$

where $\vec{x}_C = (x_C, y_C, z_C)^T$ is the current position of the camera, and q_C is the current orientation of the camera as a quaternion.

4.3.4 Landmark Initialization

When first creating a landmark from an observation, the initial landmark state must be calculated. For lidar-based systems, it is simply a matter of inverting the measurement function to produce an initial estimate for the 3D world position of the landmark. In bearing-only systems, however, the inverse measurement function does not exist. This is due to the lack of depth information from the image.

In order to initialize the landmark in the system, an external source of depth information is needed. Once ρ is known, the measurement inverse may be calculated as in Equation (42).

$$\begin{bmatrix} \theta \\ \phi \\ \rho \end{bmatrix} = h^{-1}(u, v) = \begin{bmatrix} \arcsin\left(r_z, \sqrt{r_x^2 + r_y^2 + r_z^2}\right) \\ \arctan(r_y, r_x) \end{bmatrix} \tag{42}$$

$$\vec{r} = q_C \begin{bmatrix} \frac{1}{m_x}(u - u_0) \\ \frac{1}{m_y}(v - v_0) \\ f \end{bmatrix} q_C^{-1} \tag{43}$$

where $\vec{x}_C$ is the current position of the camera to be stored as a parameter to the landmark, q_C is the current orientation of the camera as a quaternion, f is the camera focal length, m_x and m_y are scale factors that convert pixel indices into metric distances on the image plane, $\binom{u_0}{v_0}$ is the principle point in pixel coordinates, and $\binom{u}{v}$ is the image pixel coordinates of the visual feature.

One solution is to initialize all landmarks at a fixed distance from the camera, and set the depth uncertainty large enough to cover the region of expected landmark depths. In outdoor environments, however, background features such as mountains or clouds can be hundreds of kilometers away, while near-field ground features can be as close as a few centimeters. The inverse depth parametrization described previously enables a large uncertainty region to be applied to the landmark, but the ultimate performance of the system will be enhanced if a good initial estimate is provided.

To estimate the depth of a new landmark, the terrain is modeled as a single plane using the robot's current state as reference. The intersection of the new landmark with this plane is used for the initial landmark depth. If the landmark observation does not intersect with the planar approximation in front of the camera (i.e. the observation is above the horizon line), the landmark is initialized at infinity ($\rho = 0$ with the inverse depth parametrization). Equation (44) shows the calculation for the inverse depth given a feature point.

$$\rho = \frac{\vec{n} \cdot \vec{r}}{-\|\vec{r}\| \left(\vec{n} \cdot \vec{x}_C - \vec{n} \cdot \vec{x}_t\right)} \tag{44}$$

$$\vec{r} = q_C \begin{bmatrix} \frac{1}{m_x}(u - u_0) \\ \frac{1}{m_y}(v - v_0) \\ f \end{bmatrix} q_C^{-1} \tag{45}$$

$$\vec{n} = q_t \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} q_t^{-1} \tag{46}$$

where $\vec{x}_C$ is the current position of the camera, q_C is the current orientation of the camera as a quaternion, q_t is the current orientation of the robot as a quaternion, f is the camera focal length, m_x and m_y are the camera scale factors, $\binom{u_0}{v_0}$ is the camera principle point, $\binom{u}{v}$ is the visual feature point.

4.3.5 Landmark Database

During each SLAM iteration, features detected in the new frame must be matched with existing landmarks. As the size of the landmark database grows over time, real-time operation of the SLAM algorithm depends on efficient execution of nearest-neighbor queries. As a first pass, techniques for culling 3D points from consideration, such as viewable camera frustum culling, can quickly eliminate a large portion of the database from consideration [2]. In this application, where the expected rover path is piecewise straight, the landmarks that are behind the rover are unlikely to ever be viewable again. Using this insight, the database is periodically culled of landmarks that are significantly behind the camera's image plane. A plane is generated several meters behind the rover along the camera's line-of-sight. Algorithm 2 is used to determine if a given landmark is behind this plane. If so, that landmark is removed from the active set. In practice, this limits the number of landmarks that must be actively maintained, while allowing the total number of landmarks used during the traverse to increase without bound. The active database size during field trials was on the order of 500 landmarks.

Algorithm 2 Cull Landmarks Behind Camera

$p_C = CameraPosition(x_t^n)$ * Calculate camera position from the robot state
$\vec{v}_C = CameraDirection(x_t^n)$ * Calculate camera direction from the robot state
$p'_C = p_C - \alpha\vec{v}_C$ * Offset the camera position backwards by α meters
for $i = 1$ **to** I **do** * Loop over all landmarks
  $p_L = LandmarkPosition(\mu_t^i)$ * Calculate the landmark position
  $\vec{v}_L = p_L - p'_C$
  $d = p_L \cdot p'_C$ * Dot product between vectors
  if $d < 0$ **then**
    $DeleteLandmark(\mu_t^i)$ * Landmark is behind the camera; delete it
  end if
end for

4.3.6 Data Association

Within the PF SLAM framework, the error between an observed feature, z_t^j, and the associated predicted observation, $\hat{z}_t^i$, is used to correct the landmark state and to weight the plausibility of the parent particle. In the presence of multiple observations and landmarks, this assumes a method of matching the observations with the predictions, or performing data association.

In Section 4.2 data association was performed using only the visual characteristics of the extracted features, as captured by the feature descriptor. Inside the SLAM system, additional information is known in the form of the expected image-space location of each landmark. To perform data association in this context, each active landmark is projected into image-space coordinates using the measurement function in Equation (39). An image-space uncertainty covariance, S_t^i, for the landmark is also produced using the standard extended Kalman filter equations. Any new observations that fall within the 3-σ ellipse of a predicted feature location are considered for potential association. The final selection is then based on a visual feature descriptor, such as the SIFT descriptor [73]. The probability of the selected association is calculated using Equation (47). If no association is made for a specific landmark, then the landmark state remains unchanged and a fixed value of $p_{missing}$ is used for the

association probability. If an observation is not associated with an existing landmark, then a new landmark is created using Equation (42) and (44) and a fixed value of p_{new} is assigned to the association probability. This system uses values of $p_{new} = 0.05$ and $p_{missing} = 0.05$.

$$p_t^i = \left|2\pi Z_t^i\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t^j - \hat{z}_t^i)^T Z_t^{i-1}(z_t^j - \hat{z}_t^i)\right) \tag{47}$$

4.3.7 GPS Fusion

SLAM systems are incremental, with the current position estimate updated based on a change in observations. One of the fundamental issues when using any incremental localization system is drift. As the system runs, small errors accumulate, resulting in significant localization error over time. In order to remove this drift, global position information, in the form of low-accuracy GPS data has been fused with vision-based SLAM.

The robot state distribution is approximated using a particle filter, which uses sequential importance resampling (SIR) to approximate the true robot state distribution from a set of weighted samples. To incorporate additional measurements into the system, an additional weighting step is applied to each robot state particle.

The consumer-grade GPS units utilized in this system publish a position fix at 1Hz, as well as an error estimate in the form of horizontal and vertical dilution of position (DOP) values. The DOP values are scale factors applied to the inherent error characteristics of the GPS unit, derived from the current satellite geometry. A Gaussian weight function is constructed from the reported GPS position and covariance using Equation (48). This system allows particles that naturally traverse near the GPS measurement to propagate forward. Since the particle trajectory is unchanged by the weighting process, no correction to the landmark positions is needed.

$$w^n_{GPS} = |2\pi\Sigma_{GPS}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\vec{dx}^T {\Sigma_{GPS}}^{-1}\vec{dx}\right) \tag{48}$$

$$\vec{dx} = \vec{x}_R - \vec{x}_{GPS} \tag{49}$$

$$\Sigma_{GPS} = \begin{bmatrix} {(hDOP\cdot\sigma_{GPS})^2} & & 0 \\ & {(hDOP\cdot\sigma_{GPS})^2} & \\ 0 & & {(vDOP\cdot\sigma_{GPS})^2} \end{bmatrix} \tag{50}$$

where $\vec{x}_t$ is the current robot position, $\vec{x}_{GPS}$ consists of the eastings, northings, and elevation from the GPS position fix, $hDOP$ and $vDOP$ are the horizontal and vertical dilution of position values from the GPS fix quality, and σ_{GPS} is the rated accuracy of the GPS unit.

4.4 Results

Data from five of the May/2009 fields trials were selected in order to test the system in a variety of glacial terrains [133]. The captured camera images, recorded robot control values, and GPS sensor readings were used as input to the described visual SLAM system. The visual SLAM systems produces a maximally likely position estimate, as well as predicts the current position error covariance based on a linearized system model. Figure 43 shows a typical example of the localization output recorded at Site F, compared with the recorded GPS values and corresponding GPS uncertainty. As seen, the calculated localization variance is significantly smaller than the GPS uncertainty. Figure 44 shows the localization results at Site C, the most challenging of the test sites due to the large terrain variability. Again, the proposed localization method significantly outperforms GPS alone in terms of measured uncertainty. Table 6 presents a summary of the traverse experiments performed, including the GPS coordinates of each site, total distance traveled, and the average 95% confidence uncertainty for the traverse, as calculated by the localization system. As expected, the calculated positional uncertainty generally increases with the total distance traveled, but stays well below the rated 10 m accuracy obtained using only the GPS sensor.

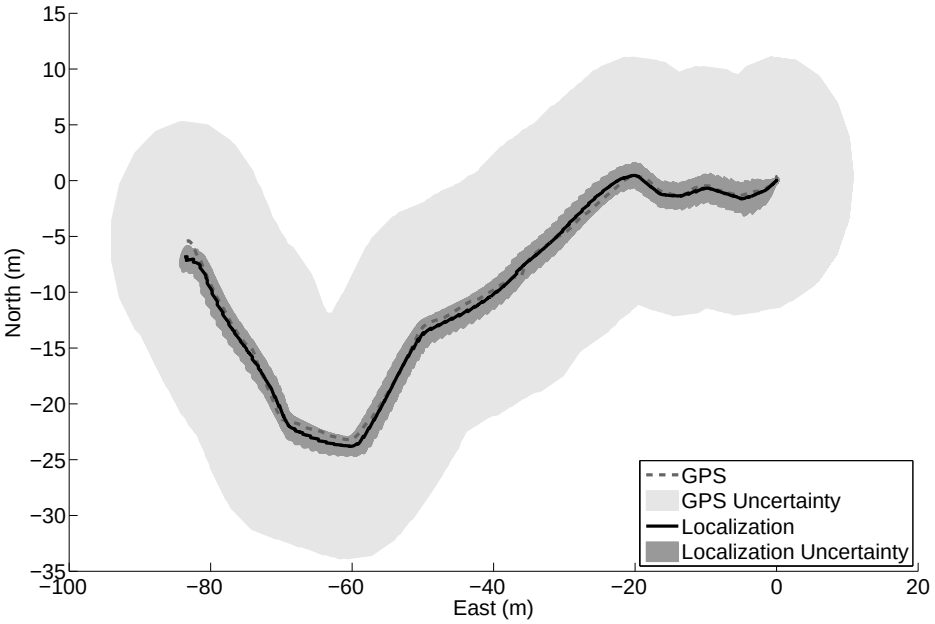


Figure 43: The localization results using GPS alone and in combination with Visual SLAM at Site F. The calculated uncertainty value has been reduced considerably through the use of vision-based techniques.

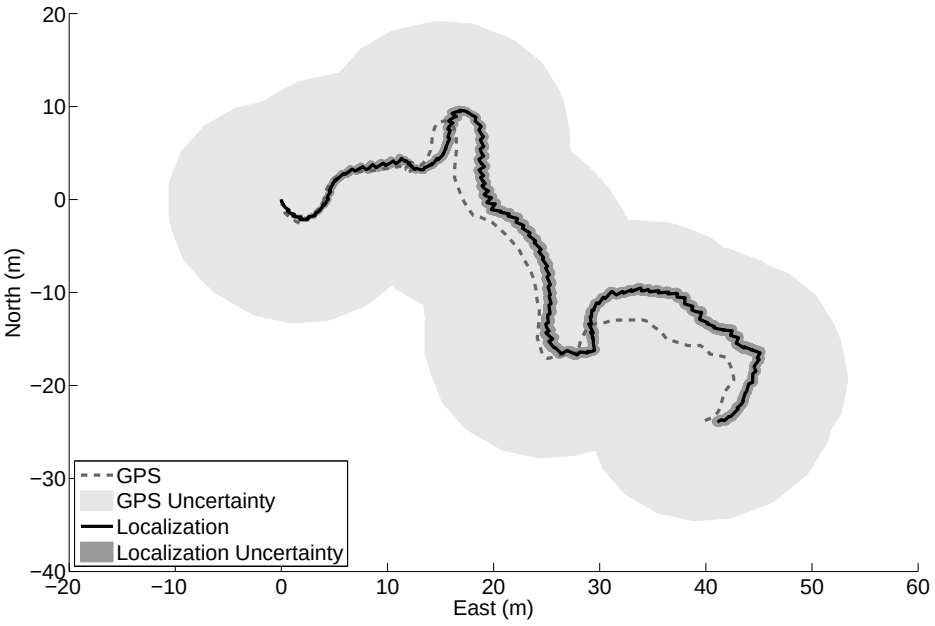


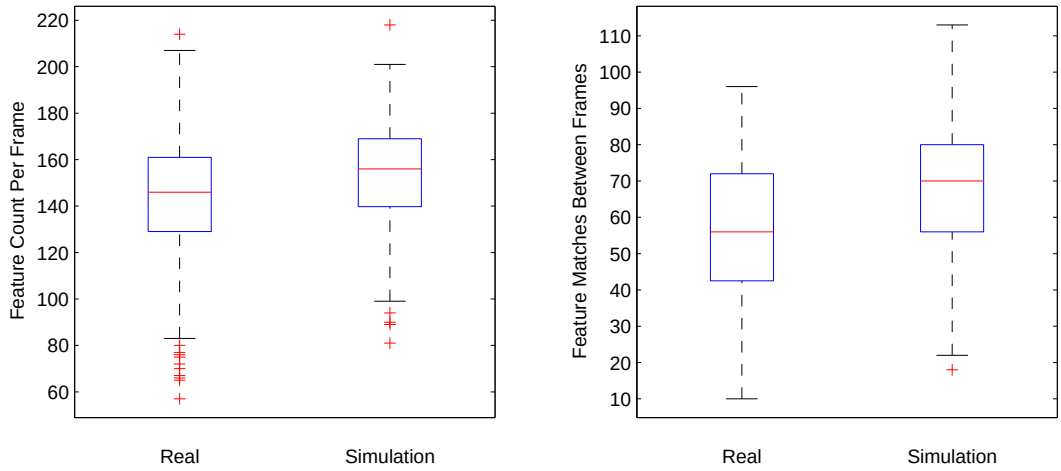
Figure 44: The localization results using GPS alone and in combination with Visual SLAM at Site C, the most challenging of the test sites due to the large terrain variability.

Table 6: Test Site Summary

Site	Reference Location	Date Collected	Distance Traveled	95% Confidence
A	58.56°N, 134.41°W	5/31/2009 10:18	38.4 m	0.59 m
C	58.47°N, 134.54°W	6/04/2009 12:34	194.2 m	1.22 m
D	58.55°N, 134.45°W	6/04/2009 14:35	180.1 m	0.97 m
E	58.55°N, 134.51°W	6/04/2009 15:24	167.0 m	0.86 m
F	58.53°N, 134.39°W	6/04/2009 17:27	100.1 m	1.64 m

The data from the field trials show the visual SLAM system is capable of operating on images of real terrain, the use of GPS data within the estimation cycle is able to reduce the drift typical of incremental localization schemes, and the uncertainty of the state estimate is significantly smaller than using GPS alone, even on long traverses. However, the data recorded from the field trials does not include a high-accuracy position or orientation measurement, and thus cannot be used to assess the absolute performance of the state estimates. Instead, the simulation system described in Chapter 2.3 has been utilized to perform a numerical evaluation of the state predictions.

Before using the simulation to validate the visual SLAM system, specific rendering quality aspects of the simulation pertinent to the localization algorithm must be investigated. The main image processing steps specific to the visual SLAM system relate to feature extraction and feature matching. To that end, two different metrics are considered in the feature extraction phase: the average number of features extracted per frame, and the average number of feature matches between frames [134]. In Figure 45 the performance of the feature extraction algorithm in the simulated environment is comparable to the performance on real images. The median of the



(a) Feature Count

(b) Feature Matches

Figure 45: Comparison of metric statistics for the visual SLAM localization algorithm calculated on images from the real environment and the simulation system. The general agreement of median and data span between the two environments indicates the simulation system is providing a good approximation of the real world.

feature matching performance tends slightly higher in simulation. Since the simulation system is uncorrupted by image noise and camera lens distortion, it is reasonable that instances of image features are more correlated between frames within the simulation system.

With an understanding of the simulation quality, the accuracy of the visual SLAM state estimates system can be investigated. The data acquired during the behavior-based control tests in Chapter 3.3 have been reused to test the performance of the visual SLAM localization system. The path executed by simulated rover performed "switchbacks" during the traverse, lengthening the total distance traveled to over 500 m. The path executed by the rover is illustrated in Figure 46. During the traverse, the simulated images, control values, and ground truth rover poses were recorded. GPS data was simulated by publishing the rover pose corrupted with zero-mean Gaussian noise at a rate of 1Hz. The 6-DOF localization state produced by the GPS-fused PF SLAM system is compared with the ground truth values in Figure 47.

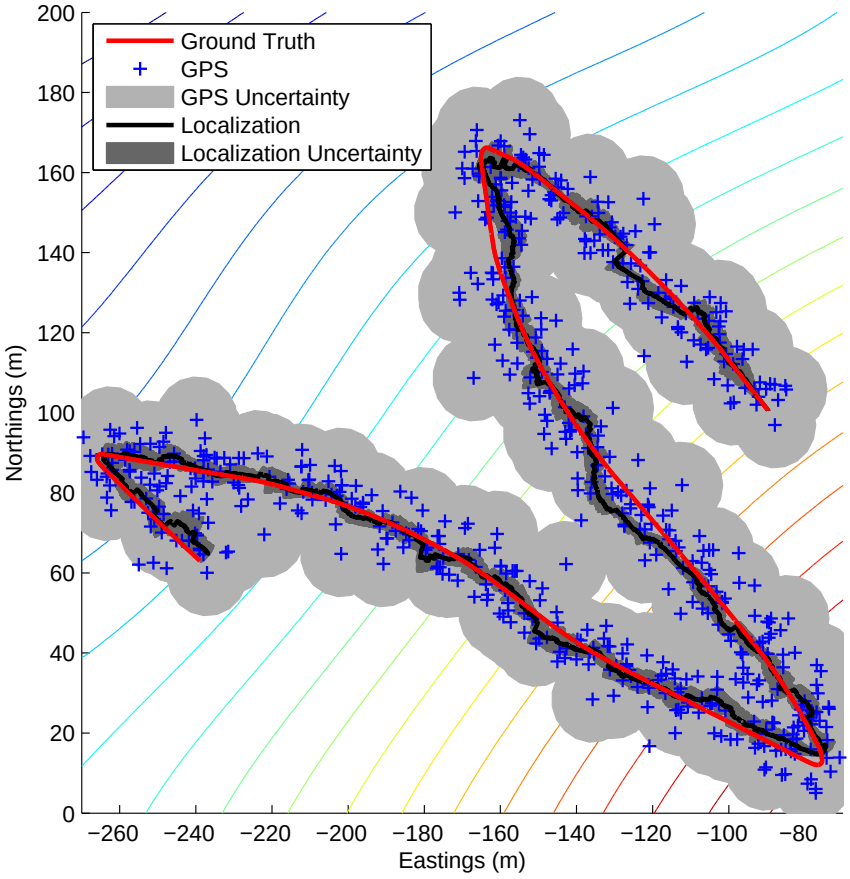
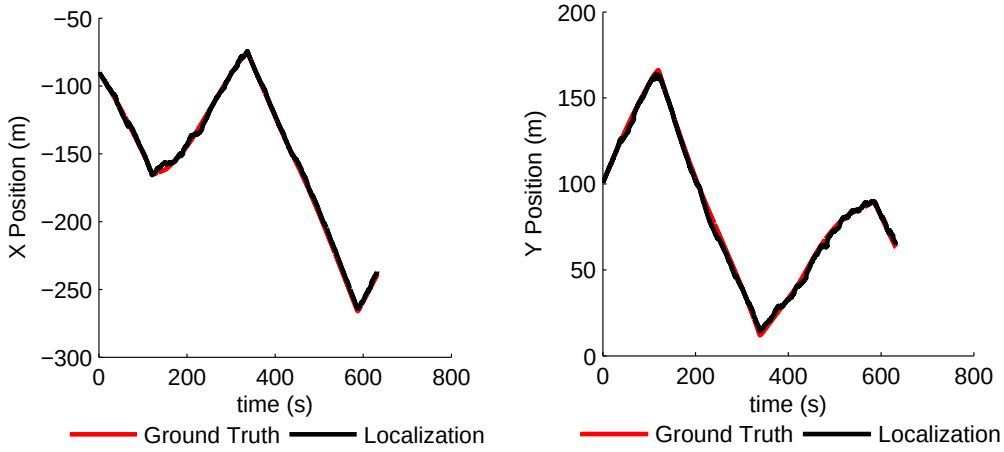


Figure 46: The path executed by the rover using a roll-minimizing control scheme, for a total path length of 537m.

Over the entire traverse, the localization system attained average absolute errors in the position terms of less than than 1.0 m, which is significantly better than the 10 m uncertainty of the GPS unit alone. This demonstrates the effectiveness of the combined system system, where the GPS prevents localization drift over large distances, while the visual SLAM component produces accurate state estimates over short distances. Also, by weighting the state particles by the GPS measurements, only those particles traveling in the proper direction will be propagated. This tends to reduce errors in pitch and roll. However, the pose estimates produced by the system are more susceptible to drift, as no source of global data is available to weight the correct orientations.

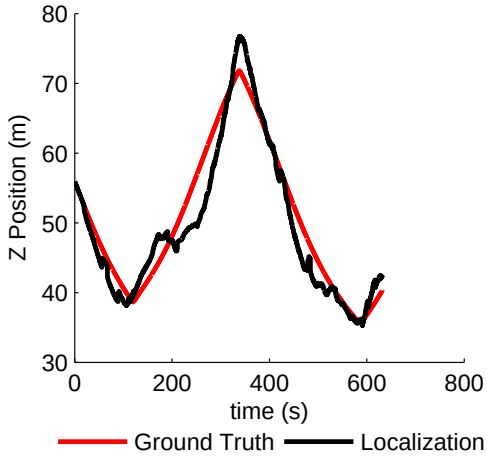
4.5 Conclusions

When developing a visual odometry system for glacial images, feature extraction is possibly the biggest challenge. Section 4.2.1 described a procedure for extracting features from these low-contrast environments. To validate this procedure, the quantity and consistency of extracted features in the raw image and enhanced image are compared. To force the comparison to concentrate on foreground, only those features that lie within the unmasked region described in Chapter 3.1 are kept for evaluation. The resulting feature sets are then matched between consecutive frames. These matches are checked against an estimate of the essential matrix to test for matching consistency. The preprocessing steps resulted in a five to ten fold increase in the number of detected features and the number of matched features when applied to three different image sequences acquired on Mendenhall Glacier in 2009. Perhaps most importantly, every enhanced image produced a minimum of ten matched features, whereas the unenhanced images often resulted in zero correctly matched features. Further, because the proposed preprocessing steps are also a requirement for the slope estimation system described in Chapter 3.2.1, no additional computation time is incurred.

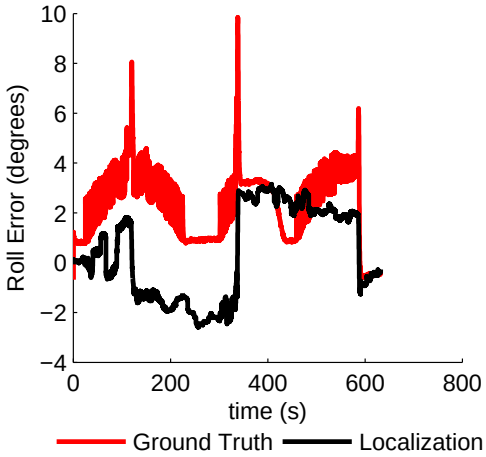


(a) Robot X Error

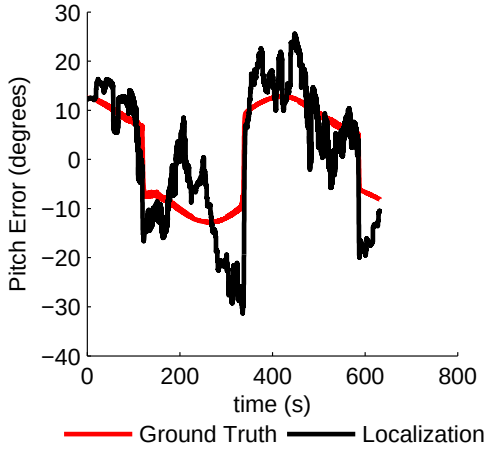
(b) Robot Y Error



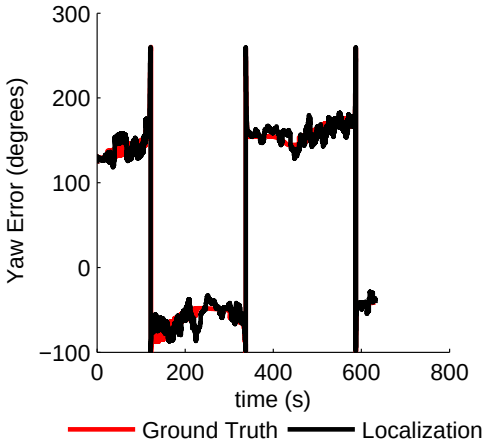
(c) Robot Z Error



(d) Robot Roll Error



(e) Robot Pitch Error



(f) Robot Yaw Error

Figure 47: Localization values compared with the ground truth values over a 537m traverse in the simulation system.

Additionally, four common feature detector algorithms were applied to the glacial image sequences, Harris, SIFT, SURF, and MSER. Again, the total number of features and total number of correct matches were compared with the enhanced image sequences as the source. The SIFT and SURF algorithms provided the best matching performance, as different image scales are employed in the extraction process. The SURF algorithm provides both the best matching performance, and has a significant computational time advantage of the SIFT detector. In practice, the SURF detector averages less than 50 ms per frame, making near real-time operation possible.

With the source of visual measurements handled, the implementation details of a vision-based PF SLAM were presented in Section 4.3, including robot state and landmark representation and GPS data fusion. The particle filter approach allowed for simple and efficient inclusion of external GPS measurements, and was able to handle large sets of active features in near real-time. This system was then tested against recorded data from field trials on Mendenhall Glacier, as well as within the simulation system. Although high-accuracy state measurements of the robot were unavailable from the field trial logs, the reconstructed paths were consistent with the GPS tracks, and provided pose estimate uncertainties on less than $1.0m$ in most cases. Utilizing the simulation system, numerical analysis of the state estimates was possible. Results showed the positional errors of the state estimate remained bounded, due to the inclusion of the GPS data, while errors in the orientation remained largely consistent with the true robot orientation. However, it is clear the addition of external orientation data, such as a consumer-grade IMU or simple inclinometer, could be beneficial to the long-term stability of the state estimates. While vision alone may be insufficient for long distance localization, the inclusion of vision data in the localization subsystem greatly enhances the accuracy, even in low-contrast glacial conditions. This allows even low-cost sensors to produce sub-meter localization results.

CHAPTER V

TERRAIN RECONSTRUCTION

Methods for terrain assessment and vision-based localization have been explored in Chapters 3 and 4. The terrain assessment has been utilized as part of a reactive control scheme, ensuring self preservation, while the localization system allows sensor measurements acquired in a local robot coordinate frame to be properly registered in a global frame for purposes of mapping or scientific data collection.

However, as the number of impassible obstacles increases, the performance of the reactive system declines. In these situations, a path planner with knowledge of all past encountered terrain is essential. While the algorithms employed by global path planners differ significantly, from the dynamic programming methods of Dijkstra's algorithm [29] to the random sampling methods of rapidly-exploring random trees [64], all planning strategies require a map on which to plan. As discussed in Chapter 3, the major obstacles in glacial environments are slope based, and asymmetries in the rover's design make knowledge of the rover's orientation in the environment important when determining traversability. Hence, a topographic map is a natural choice, allowing the planning algorithm to predict the rover orientation over the entire path.

A terrain reconstruction method is presented in the following sections that creates a topographic terrain map. This method uses the sparse landmark position estimates from the localization system, and combines it with the dense slope estimates from the terrain assessment using a statistical construct known as a Gaussian process (GP). A GP intrinsically handles measurement uncertainty, allowing the calculation of both the maximally likely terrain surface, and the terrain uncertainty at any point. It is shown that the GP framework is capable of generating a reasonable terrain model

using only the sparse landmark points, outperforming the standard triangular mesh interpolation, particularly at large distances. Further, it is shown that the incorporation of slope information into the GP significantly improves the reconstruction, something not easily integrated into simpler interpolation schemes.

5.1 Gaussian Processes

A Gaussian process (GP) is a collection of an infinite number of random variables with a jointly Gaussian distribution [95]. This may be interpreted as a distribution over continuous functions, similar to how a Gaussian variable defines a distribution over real values. Instead of sampling a value in $\mathbb{R}^N$ from the Gaussian variable, a continuous function, $f(\vec{x})$, is drawn from the GP that maps an input vector, $\vec{x} \in \mathbb{R}^N$, to an output value, $y \in \mathbb{R}$. A GP is defined by a mean function, $\mu(\vec{x})$, which describes the mean output value of all possible sample functions evaluated at the input, $\vec{x}$, and a covariance function, $k(f(\vec{x}_i), f(\vec{x}_j))$, which describes the correlation between any pair of output values. The choice of the mean and covariance functions allows prior knowledge of the function's behavior to be encoded in the GP framework. While many covariance functions are possible, a common and natural choice is the squared exponential function listed in Equation (51). This covariance function is derived from a Gaussian kernel, exhibits rotation and translation invariance to the inputs, and is infinitely differentiable or infinitely smooth. The exact spatial behavior of the covariance function can be tuned with a function-dependent set of parameters, known as hyperparameters in GP literature [95].

$$k\left(f(\vec{x}_i), f(\vec{x}_j)\right) = \alpha \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{x}_j\right)^T \Gamma \left(\vec{x}_i - \vec{x}_j\right)\right) \tag{51}$$

where Γ is a diagonal matrix of elements $\frac{1}{\gamma_1}, \ldots, \frac{1}{\gamma_N}$, and α is a scaling factor. The variables in the $N+1$ dimensional set $\alpha, \gamma_1, \ldots, \gamma_N$ are known as the hyperparameters for the squared exponential Gaussian process.

To draw a sample, the GP must be evaluated at each input value, $\vec{x}$. However, as stated previously, a GP is an infinite dimensional object. Despite the infinite dimensional nature of GPs, sampling is still computationally tractable due to the marginalization property. If a GP is defined over a set, S, by $\mathcal{GP}(\mu, \Sigma)$, then the GP is also defined over any subset of S by the relevant submatrices of μ and Σ, as shown in Equation (52). Thus, as long as the number points at which $f(\vec{x})$ is to be evaluated is finite, then sampling from the GP is also finite. Equivalently, any finite set of variables from a GP have a jointly Gaussian distribution [95]. An example of several functions drawn from a GP prior with zero mean and squared exponential covariance function are shown in Figure 48a.

$$p(y_i, y_j) \sim \mathcal{N}\left(\begin{bmatrix}\mu_i \\ \mu_j\end{bmatrix}, \begin{bmatrix}\Sigma_{i,i} & \Sigma_{i,j} \\ \Sigma_{j,i} & \Sigma_{j,j}\end{bmatrix}\right) \Rightarrow p(y_i) \sim \mathcal{N}\left(\mu_i, \Sigma_{i,i}\right) \tag{52}$$

A GP can also be conditioned on a set of known measurements [95]. The resulting GP posterior describes only the subset of sample functions that pass through the measurement points. This allows the GP to be used as a regression or interpolation technique, in which samples may be queried at an arbitrarily small resolution. However, unlike conventional regression or interpolation techniques, no data model (linear, quadratic, etc.) is required. For interpolation, a set of unknown output values, $Y^* = \left\{y_j^* | j = 1, \ldots, Q\right\}$, is desired, corresponding to a set of known inputs values, $X^* = \left\{\vec{x}_j^*\right\}$. The output values are to be conditioned on a set of known measurements, $Y = \{y_i | i = 1, \ldots, P\}$, corresponding to a second set of known input values, $X = \{\vec{x}_i\}$. The GP posterior mean and covariance satisfying these conditions are shown in Equation (54) and (55) (with a full derivation available in [122]). An illustration of the resulting posterior is shown in Figure 48b.

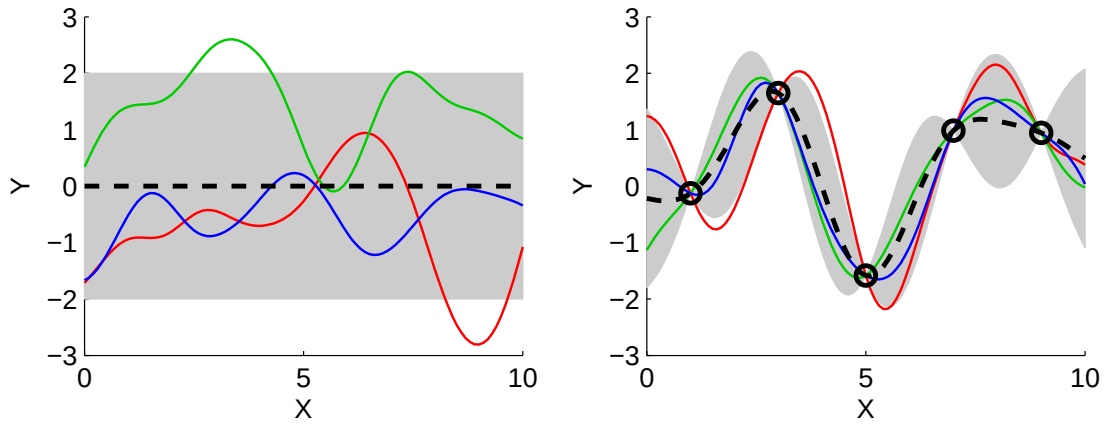


Figure 48: (a) An example of three functions sampled from a zero mean, unit variance Gaussian process, prior to applying any conditions. The mean is shown in black, and the 95% confidence area is shaded in gray. (b) An example of three functions sampled from a Gaussian process after conditioning on five measurement values. Again, the mean function is shown in black, and the 95% region has been shaded in gray.

$$p(Y^*|X, Y, X^*) \sim \mathcal{N}(\mu^*, \Sigma^*) \tag{53}$$

$$\mu^* = \mu_X + \Sigma_{Y,Y^*} \cdot \Sigma^{-1}_{Y^*,Y^*} \cdot (Y - \mu_X) \tag{54}$$

$$\Sigma^* = \Sigma_{Y,Y} - \Sigma_{Y,Y^*} \cdot \Sigma^{-1}_{Y^*,Y^*} \cdot \Sigma^T_{Y,Y^*} \tag{55}$$

where μ_S is a vector of values produced by evaluating the mean function, $\mu(\cdot)$, over the set, S, and Σ_{S_1,S_2} is a covariance matrix constructed by evaluating the covariance function, $k(\cdot,\cdot)$, with each pair-wise combination of values from sets S_1 and S_2.

The GP framework also allows the incorporation of derivative information, either as query points to be returned, or as measurements to be used as conditions [106]. An output value, ω_j^m, is defined to be the partial derivative of the output function, $f(\cdot)$, with respect to the m^{th} dimension of the input, evaluated at the input value, $\vec{x}_j$, as shown in Equation (56).

$$\omega_j^m = \frac{\partial f(\vec{x})}{\partial x^m}|_{\vec{x}=\vec{x}_j} \tag{56}$$

The joint probability of the mixed vector containing y_i and ω_j^m involves evaluating the mean function, $\mu(\cdot)$, and covariance function, $k(\cdot,\cdot)$. If the mixed vector is used to condition the GP, then the mean values will come directly from the measured values. If the mixed vector contains query points, then the mean values will be derived from Equation (53). The covariance function, on the other hand, must be evaluated for all possible pairs contained inside the mixed vector for both cases. Equations (57) and (58) show how first derivative information may be incorporated into the existing problem structure, assuming the covariance function is differentiable [106].

$$k\left(\omega_i^m, y_j\right) = \frac{\partial}{\partial x^m} k\left(f(\vec{x}_i), f(\vec{x}_j)\right) \tag{57}$$

$$k\left(\omega_i^m, \omega_j^n\right) = \frac{\partial^2}{\partial x^m \partial x^n} k\left(f(\vec{x}_i), f(\vec{x}_j)\right) \tag{58}$$

Specifically, for the squared exponential covariance function in Equation (51), the possible modified covariance functions are listed in Equations (59) - (61).

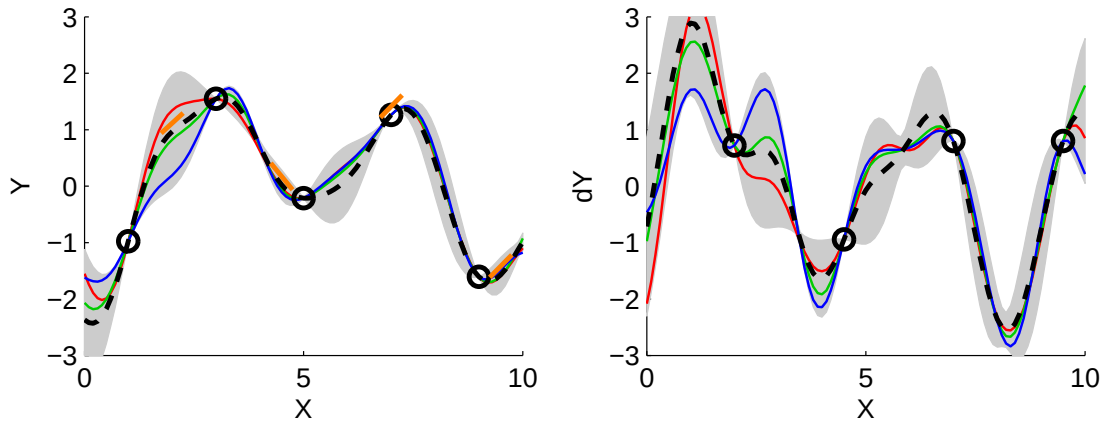
$$k\left(\omega_i^m, y_j\right) = -\frac{1}{\gamma_i}\left(x_i^m - x_j^m\right)\alpha \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{x}_j\right)^T \Gamma\left(\vec{x}_i - \vec{x}_j\right)\right) \tag{59}$$

$$k\left(y_i, \omega_j^m\right) = \frac{1}{\gamma_i}\left(x_i^m - x_j^m\right)\alpha \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{x}_j\right)^T \Gamma\left(\vec{x}_i - \vec{x}_j\right)\right) \tag{60}$$

$$\begin{aligned} k\left(\omega_i^m, \omega_j^n\right) = & -\frac{1}{\gamma_i}\frac{1}{\gamma_j}\left(x_i^m - x_j^m\right)\left(x_i^n - x_j^n\right)\alpha \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{x}_j\right)^T \Gamma\left(\vec{x}_i - \vec{x}_j\right)\right) \\ & + \delta_{m,n}\frac{1}{\gamma_i}\alpha \exp\left(-\frac{1}{2}\left(\vec{x}_i - \vec{x}_j\right)^T \Gamma\left(\vec{x}_i - \vec{x}_j\right)\right) \end{aligned} \tag{61}$$

where $\delta_{m,n}$ is the Kronecker delta function, which is one when $m = n$ and zero otherwise.

A 1D example is shown in Figure 49. In this example, five evenly spaced position constraints were selected at random, along with four derivative constraints at a different spacing. The x-position of each measurement are stacked into the column vector, X, of length 9. The measurement values are similarly stacked into the column vector, Y. Y consists of both direct measurements, y_i, and derivative measurements,



(a) (b)

Figure 49: An example of three functions sampled from a Gaussian process after conditioning on five measurement values and four derivative values. The mean function is shown in black, and the 95% region has been shaded in gray. Both the function output (a) and the function derivative (b) are shown. Derivative constraints are indicated in the position plot as short, orange lines aligned in the direction of the derivative.

ω_i^1. Interpolated values for the function and the first derivative are requested at a resolution of 0.1 over the range $[0, 10]$. The 101 x-positions for the function output are stacked with the 101 x-positions for the derivative output to form the vector, X^*. The mean and covariance for the GP posterior are calculated using Equations (54) and (55), where the individual entries in the covariance matrices $\Sigma_{Y,Y}$, Σ_{Y,Y^*}, and Σ_{Y^*,Y^*} are calculated using Equations (51) and (59) - (61) as appropriate to the type of input. Functions can then be drawn from the GP by sampling from the 202-dimension multivariate Gaussian variable, $\mathcal{N}(\mu^*, \Sigma^*)$ defined in Equations (54) - (55).

5.2 Vision-based Terrain Reconstruction

One of many applications of Gaussian processes is in the field of geostatistical terrain modeling. A common procedure, known as "Kriging", involves using a Gaussian process with a special covariance function [112]. Existing terrain survey data is used to generate a variogram, which describes the terrain variation as a function of spatial distance. One of several common models is then fitted to the variogram plot, and this

model is used as the covariance function in a 2D GP interpolation. More recently, efforts at the University of Sydney have shown improved terrain modeling performance with GPs that use neural network-inspired covariance functions [118,119]. The neural network covariance function is nonstationary, allowing it to better adapt to terrains with significant discontinuities. This work successfully combined data from a sparse GPS survey with two high resolution 3D laser scans into a single terrain model, illustrating the ability of GPs to be applied to large-scale problems.

The geostatistical modeling example uses data collected specifically for the purposes of reconstructing the terrain in an off-line process. GPS survey points were distributed across the entire target region, while the laser scan locations were deliberately positioned. While this is desirable in the domain of terrain modeling, the needs of mobile field robotics dictate the topographic map be generated using only the limited data acquired at a given point in time. The following sections describe the implementation of a 2D GP terrain model that incorporates only the limited terrain data acquired during the robot's normal traverse.

5.2.1 Visual Landmarks

A Gaussian Process terrain model is capable of combining multiple measurements of the terrain elevation into a single, cohesive representation. The GP model also incorporates any measurement uncertainty into the reconstruction, if that uncertainty may be modeled by additive independent Gaussian noise. In that case, the measurement covariance matrix, $\Sigma_{Y,Y}$, is simply augmented by the elevation uncertainty of each measurement, as in Equation (62). Geostatistical models often use GPS survey data collected in a uniform grid or other sampling technique designed to capture the observed terrain variation. The positional errors associated with GPS survey data tend to be small and relatively uncorrelated, making this a good fit for GP interpolation.

$$\Sigma'_{Y,Y} = \Sigma_{Y,Y} + \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_N^2 \end{bmatrix} \tag{62}$$

where σ_i^2 is the elevation variance of the i^{th} measurement.

The visual SLAM system described in Chapter 4 produces a set of 3D point estimates that lie on the terrain surface as a byproduct of the localization process, superficially analogous to GPS data. However, unlike GPS surveys, this data is collected opportunistically while the robot performed a traverse, rather than with the explicit goal of capturing terrain variations. These visual landmarks also cover the terrain only sparsely, with landmarks near the rover's path occurring far more frequently than landmarks at significant distances. While this may be suboptimal from a terrain sampling standpoint, no additional travel is incurred by the rover to collect this data.

Further, the uncertainty of each SLAM landmark is a jointly Gaussian distribution in both the dependent variables, (x, y), and the independent variable, z. Inclusion of uncertainty in the dependent variables is known as the "error-in-variables" problem in statistics, and few solutions exist for the multi-variate case [43]. Rather than attempting to modify the GP structure to incorporate "error-in-variables" uncertainty, each landmark covariance is converted into independent additive noise by marginalizing out the dependent variables, (x, y), from the joint distribution, as shown in Equation (63). Due to the highly directional nature of visual SLAM landmark estimates, removing the dependency of x and y, even from covariances with even a small volume, results in a large elevation uncertainty. Figure 50 illustrates this behavior with a simple 2D example. For this reason, only those landmark estimates whose depth uncertainty have collapsed to a small region are considered for inclusion in the GP terrain reconstruction.

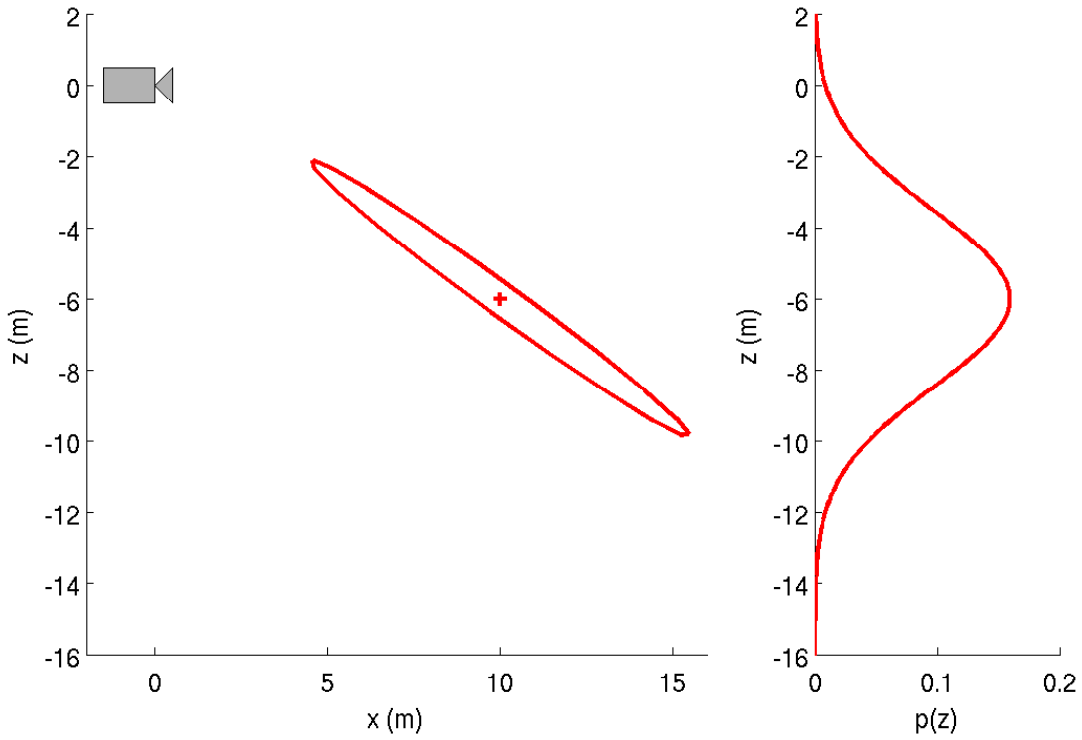


Figure 50: A simple 2D example of the uncertainty ellipse of a visual SLAM landmark, and the elevation uncertainty obtained through marginalization.

$$
\begin{aligned}
p(z) &= \int_x \int_y p(x, y, z)\, dx\, dy \\
&= \int_x \int_y \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu y \\ \mu_z \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & \alpha\sigma_x\sigma_y & \beta\sigma_x\sigma_z \\ \alpha\sigma_y\sigma_x & \sigma_y^2 & \gamma\sigma_y\sigma_z \\ \beta\sigma_z\sigma x & \gamma\sigma_z\sigma_y & \sigma_z^2 \end{bmatrix} \right) dx\, dy \\
&= \mathcal{N}\left(\mu_z, \sigma_z^2\right)
\end{aligned}
\tag{63}
$$

5.2.2 Visual Slope Estimates

As previously mentioned, the visual SLAM landmark data provides only sparse terrain information, sampled at suboptimal points from a terrain reconstruction standpoint. Augmenting these sparse elevation estimates with dense estimates of the elevation derivative improves the overall reconstruction quality. The slope estimate subsystem, presented in Chapter 3.2, provides the terrain slope, or elevation derivative, within the camera coordinate frame. To include the slope estimates in the GP model, the

slopes must be projected into the world coordinate system. A planar patch is derived from each slope estimate by rotating a plane aligned with the camera orientation by the estimated slope. This coordinate transformation is shown in Equations (64) - (66). Once the plane equation is created, the world coordinate derivatives may be calculated, as in Equations (68) and (69). The uncertainty of the slope measurements may also be incorporated into the GP model. The results presented in Chapter 3.2 provide an estimate of the slope measurement variance. This variance is projected into global coordinates using the Jacobian of the projection equations.

$$n_x x + n_y y + n_z z + (\vec{n} \cdot \vec{p}) = 0 \tag{64}$$

$$\vec{n} = q_C \; q_{slope} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} q_{slope}^{-1} \; q_C^{-1} \tag{65}$$

$$q_{slope} = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) \\ 0 \\ 0 \\ sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{66}$$

where θ is the estimate slope angle, q_C is the camera orientation as a quaternion, $\vec{p}$ is any point on the planar patch, $\vec{n}$ is the normal to the planar patch, and the vector $\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$ is the world up vector in camera coordinates.

$$z = \frac{1}{n_z} \left(-n_x x - n_y y + \vec{n} \cdot \vec{p}\right) \tag{67}$$

$$\frac{\partial z}{\partial x} = \frac{-n_x}{n_z} \tag{68}$$

$$\frac{\partial z}{\partial y} = \frac{-n_y}{n_z} \tag{69}$$

Finally, the GP requires the derivative measurements be applied at known values of the dependent variables, (x, y), just like the elevation measurements. However, unlike the landmark position estimates, the visual slope estimates do not provide any source of global localization. If a terrain model is available from a previous terrain reconstruction, it is possible to calculate the intersection point of the slope estimate

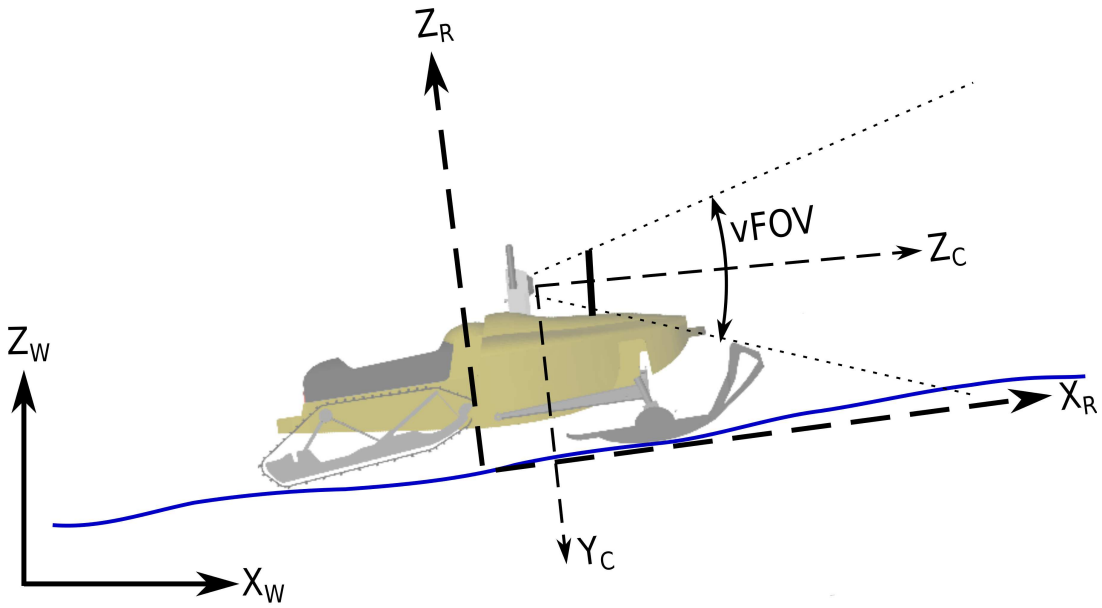


Figure 51: An illustration of the relevant SnoMote geometry used in the image pixel projection equations.

pixel with this terrain model. However, this ray-casting procedure is a numerically expensive operation, which is often approximated even inside the computer graphics community [40]. A courser approximation can be implemented at little computational cost by assuming the area in the vicinity of the robot is planar. Given the geometry of the rover and the current pose estimate from the localization system, each image pixel can be projected onto the assumed ground plane. The projection based on the robot geometry can actually be precomputed, requiring only the transformation from the local robot frame into the global frame at run-time. Figure 51 shows the construction geometry of the SnoMote, with the relevant projection shown in Equation (70).

$$\vec{n} = q_t \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} q_t^{-1} \tag{70}$$

$$\vec{r} = q_C \begin{bmatrix} \frac{1}{m_x}(u - u_0) \\ \frac{1}{m_y}(v - v_0) \\ f \end{bmatrix} q_C^{-1} \tag{71}$$

$$d = \frac{-\|\vec{r}\|\left(\vec{n} \cdot \vec{x}_C - \vec{n} \cdot \vec{x}_t\right)}{\vec{n} \cdot \vec{r}} \tag{72}$$

$$\vec{x}_P = \vec{x}_C + d\,\vec{r} \tag{73}$$

where $\vec{x}_P$ is the world coordinate position estimate of the pixel $\left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right)$, $\vec{x}_C$ is the current position of the camera, q_C is the current orientation of the camera as a quaternion, q_t is the current orientation of the robot as a quaternion, f is the camera focal length, m_x and m_y are the camera scale factors, and $\left(\begin{smallmatrix} u_0 \\ v_0 \end{smallmatrix}\right)$ is the camera principle point.

5.2.3 Hyperparameter Optimization

The GP framework is considered a model-free regression technique in that no functional model, such as a linear or logarithmic function, is used during the data fit. However, the behavior of the GP can be tuned to a specific problem through the use of the covariance function hyperparameters. In Equation (51), the terms $\alpha, \gamma_1, \ldots, \gamma_N$ are known as the hyperparameters for the squared exponential Gaussian process. The values γ_i are often referred to as the length scales. The distance between the input variables in each dimension is divided by the corresponding length scale value during the covariance calculation. This allows the GP to vary how quickly the output can change in response to the inputs. In terrain modeling, length scales in the tens to hundreds of meters are common. The α parameter is a gain value placed on the entire covariance function. This allows the GP model to be more or less sensitive to the input values as a whole.

The hyperparameters for a GP model are ideally trained on a subset of data to

maximize the posterior probability shown in Equation (74). However, if no prior probability information is known for the hyperparameter distribution, $p(\theta)$, then the common practice of maximizing the log marginal likelihood, $log\left(p(Y|X,\theta)\right)$, is equivalent. The log marginal likelihood for a GP is shown in Equation (75) [94].

$$p\left(\theta|X,Y\right) = \frac{p\left(Y|X,\theta\right)\ p\left(\theta\right)}{p\left(Y|X\right)} \tag{74}$$

$$\log\left(p\left(Y|X,\theta\right)\right) = -\frac{1}{2}Y^T\,\Sigma_{Y,Y}^{-1}\,Y - \frac{1}{2}\log|\Sigma_{Y,Y}| - \frac{n}{2}\log 2\pi \tag{75}$$

To train the hyperparameters, the locations, X, and elevations, Y, of a small segment of the simulation environment was provided to the GP. The values of the hyperparameters α and γ were varied over a large range, and the corresponding terrain reconstruction error was calculated from the ground truth elevation data provided by the simulation. Since the orientation of the world coordinate system should not effect the GP results, the length scales in the two dependent variables are set equal, $\gamma_x = \gamma_y = \gamma$. The resulting average prediction error of the GP versus the two hyperparameters is shown as a contour plot in Figure 52. The maximum point of $\alpha = 10.0$ and $\gamma = 315.0$ was selected for use in the GP regression in all following results.

5.3 Results

The data acquired during the visual SLAM simulation tests in Chapter 4.4 have been reused to test the performance of the GP terrain reconstruction system. While the visual SLAM filter estimates the position of surface landmarks as a byproduct of robot localization, each visual SLAM particle inside the filter maintains its own map of landmarks. For the purposes of terrain reconstruction, only the landmarks from the highest-weighted particle are considered. Also, the SLAM system periodically purges landmarks from the map after they are no longer visible. During the purge process,

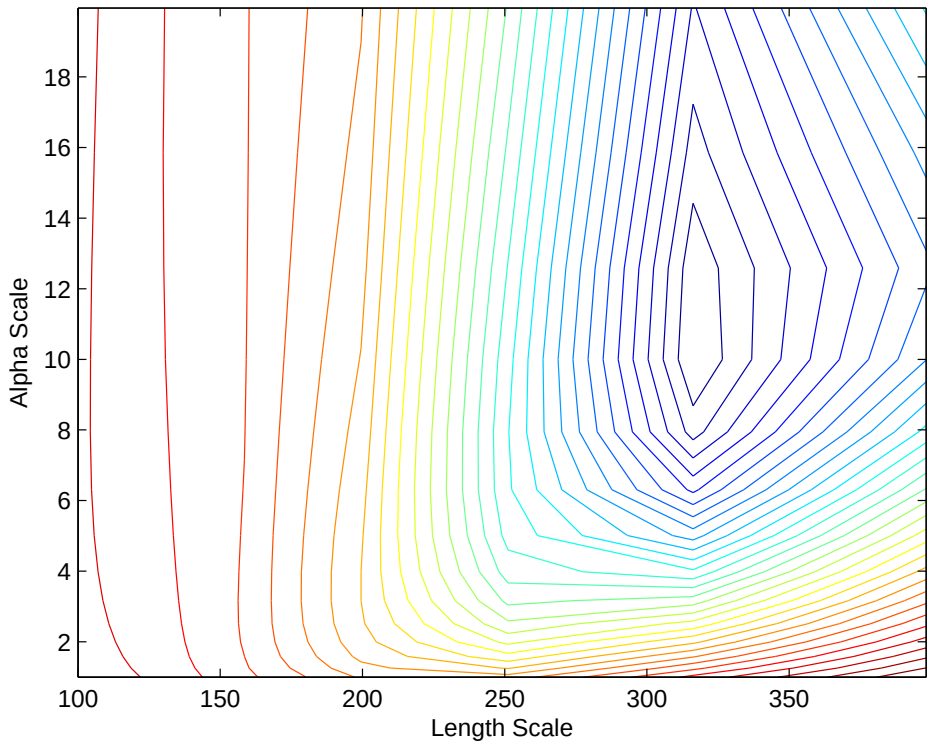


Figure 52: The contour plot of the GP reconstruction error versus the hyperparameters α and γ.

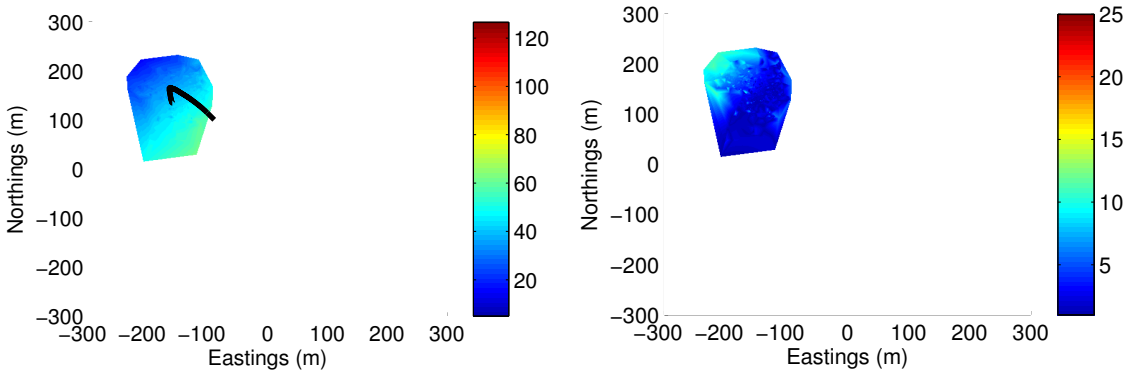
any landmark whose 1-σ covariance ellipse was no larger than $5m$ was copied to an external array. During the simulation trial, approximately 150,000 surface landmarks were sufficiently localized, a vast majority of which occurred very near the rover's path. Because of the proximity of these landmarks, the information they provide is largely redundant. To reduce the number of measurements that must be processed by the GP, only those landmarks that were initialized more than $20m$ from the rover's position are used within the reconstruction. This reduces the set to approximately 5,000 landmarks over the $600m$ x $600m$ simulation site.

Additionally, the terrain slope estimates described in Chapter 3.2 were calculated on the incoming camera stream at a rate of 1Hz. These slope estimates have been projected into world coordinates using Equations (64) - (69) based on the camera pose of the highest-weighted particle. As the dense slope estimate system provides slope information for every terrain pixel in the image, this equates to approximately 95 million individual slope estimates. To reduce the amount of slope data to be processed by the GP, the slope estimates have been resampled on a $5m$ grid. All slope estimates that exist within a $2.5m$ radius of each grid point are averaged together into a single value. This reduces the total number of slope estimates to, at worst, $14,600$ slope measurements in each of the X and Y directions.

To compare the performance of the terrain reconstruction system, three different methods are tested. The first uses a simple linear triangular mesh interpolation method. The Delaunay triangulation [25] is first formed from the input positions. Each query point that falls within the triangulation is estimated using the plane formed by the triangle's vertices. Because query points must fall within a Delaunay triangle to be estimated, this method only produces terrain estimates within the convex hull of the input measurements. Also, there is no obvious mechanism for incorporating measurement uncertainty or terrain derivatives into a triangular mesh model.

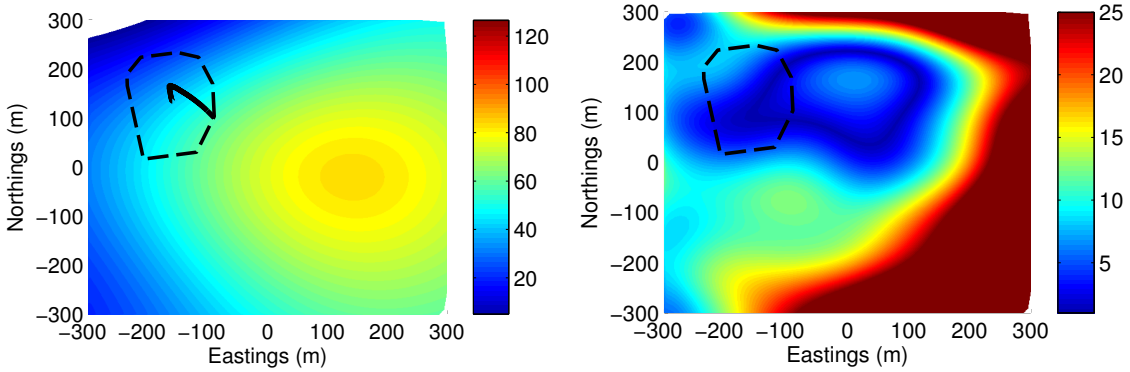
The second reconstruction incorporates the sparse visual SLAM landmark data into a Gaussian Process model. Unlike the triangular mesh interpolation scheme, the GP model is valid over all of $\mathbb{R}^2$. The final reconstruction uses a GP model that incorporates both the sparse visual SLAM landmark data and the dense terrain slope estimates. As the goal of the terrain reconstruction is to provide a path planner with a topographical map as early as possible, the performance of each system is compared with different levels of input data. In each of the four reconstruction tests, only the data collected during the first 25%, 50%, 75%, and 100% of the rover path are provided to the reconstruction models. The resulting terrain reconstructions of the $600m$ x $600m$ simulation environment are shown in Figures 53-56, with the ground truth elevation map shown in Figure 57 for reference.

Perhaps the most striking aspect of the three reconstructions is the limited data provided by the triangular mesh. During the first 25% of the traverse, only 6.1% of terrain could be reconstructed, while only 25.8% could be reconstructed after the traverse was completed. In contrast, both GP reconstructions are able to predict the elevation of the entire terrain based on the local observations, even terrain sections located in the northeast, which were behind the rover over the entire traverse. Using just the visual SLAM landmarks, the terrain model is able to predict the existence of a large hill in southeast as early as half way through the traverse. By the end of the traverse, the general S-shape of the terrain has been successfully recovered. Once the terrain slope information is introduced, the GP reconstruction error reduces considerably. With only 25% of the traverse complete, significant reconstruction errors exist only at the far edges of terrain. The root mean squared (RMS) elevation errors of each reconstruction are summarized in Table 7. Because of the limitations of the triangular mesh method, two sets of errors are calculated for each reconstruction: the first considers just the terrain within the convex hull of the input points, the second considers the entire $600m$ x $600m$ simulation environment. Within the convex



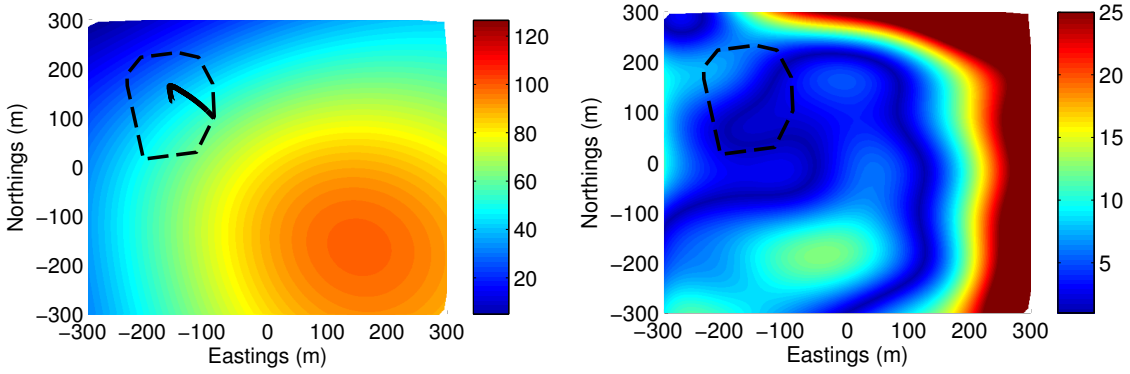
(a) Triangular Mesh Reconstructed

(b) Triangular Mesh Error



(c) GP Landmark Reconstructed

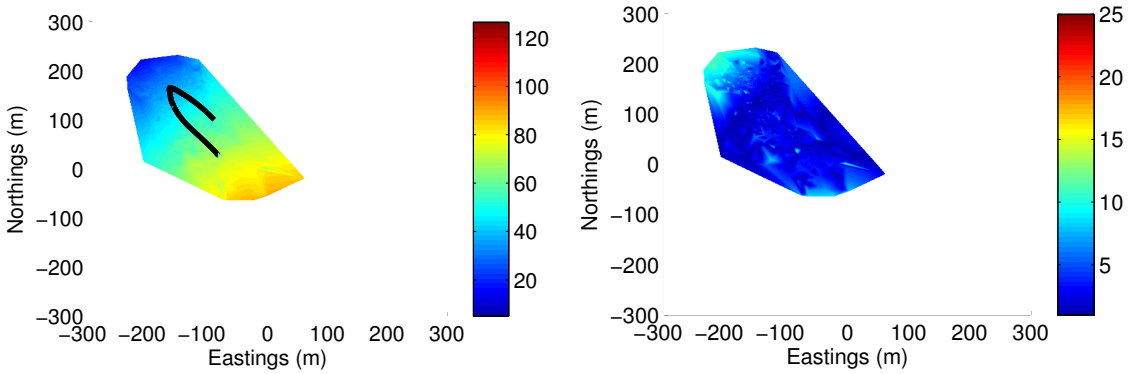
(d) GP Landmark Error



(e) GP Landmark and Slope Reconstructed

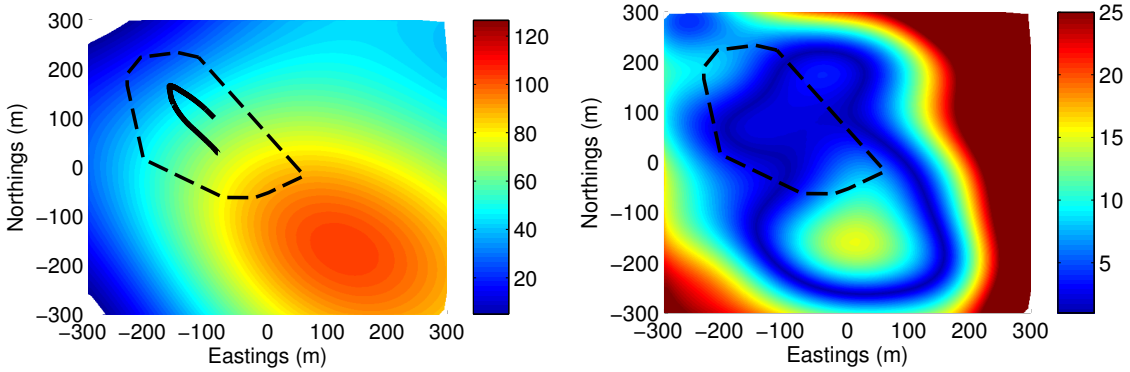
(f) GP Landmark and Slope Error

Figure 53: Terrain reconstructions using data from the first 25% of the traverse. The rover's path is shown as a solid black line, while the convex hull of landmark points is indicated by a dashed line.



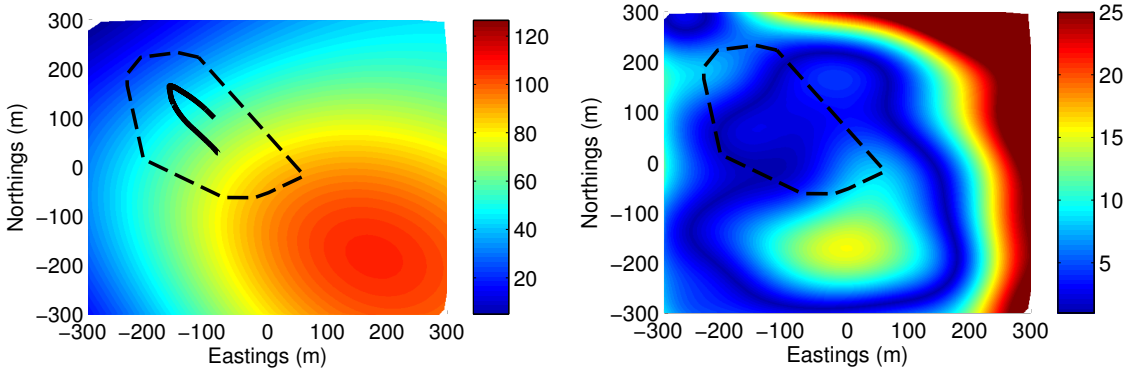
(a) Triangular Mesh Reconstructed

(b) Triangular Mesh Error



(c) GP Landmark Reconstructed

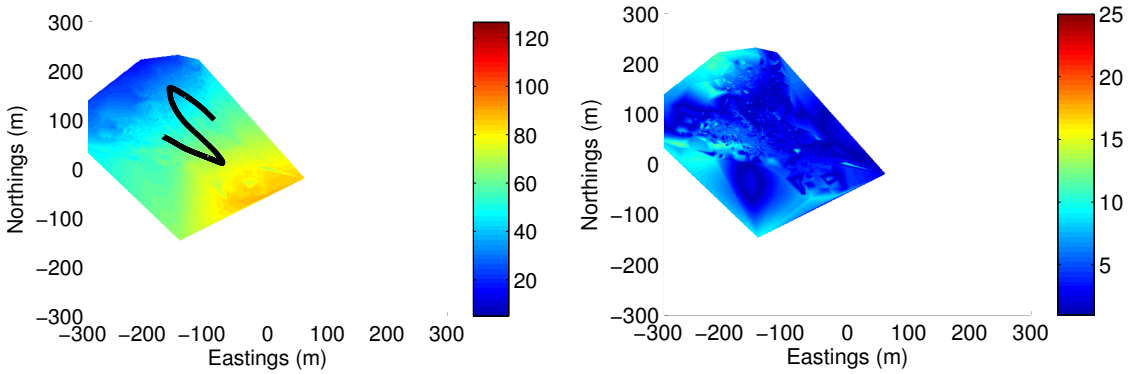
(d) GP Landmark Error



(e) GP Landmark and Slope Reconstructed

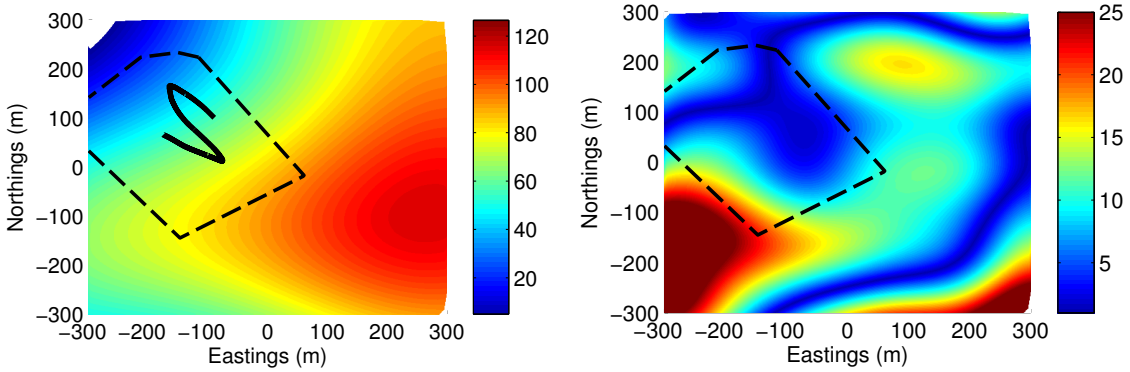
(f) GP Landmark and Slope Error

Figure 54: Terrain reconstructions using data from the first 50% of the traverse. The rover's path is shown as a solid black line, while the convex hull of landmark points is indicated by a dashed line.



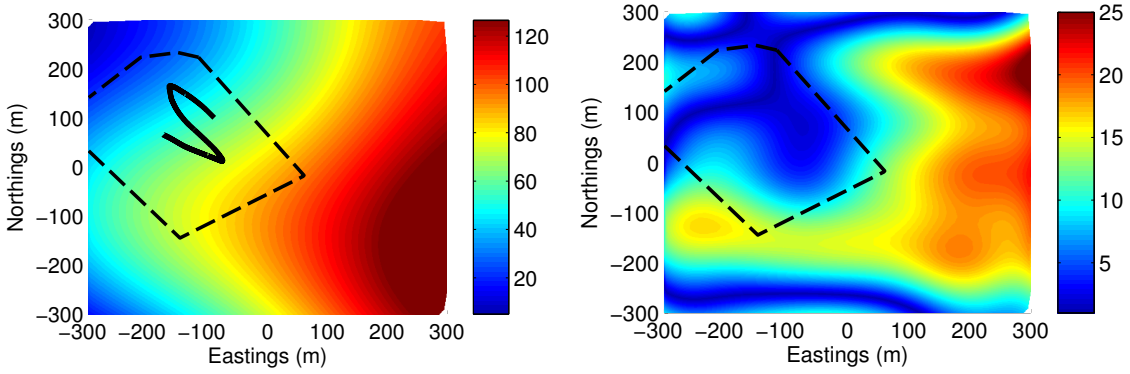
(a) Triangular Mesh Reconstructed

(b) Triangular Mesh Error



(c) GP Landmark Reconstructed

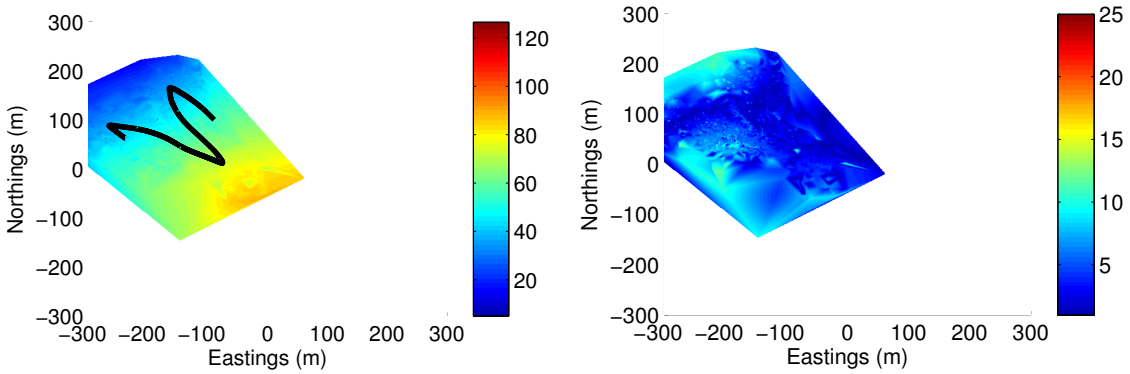
(d) GP Landmark Error



(e) GP Landmark and Slope Reconstructed

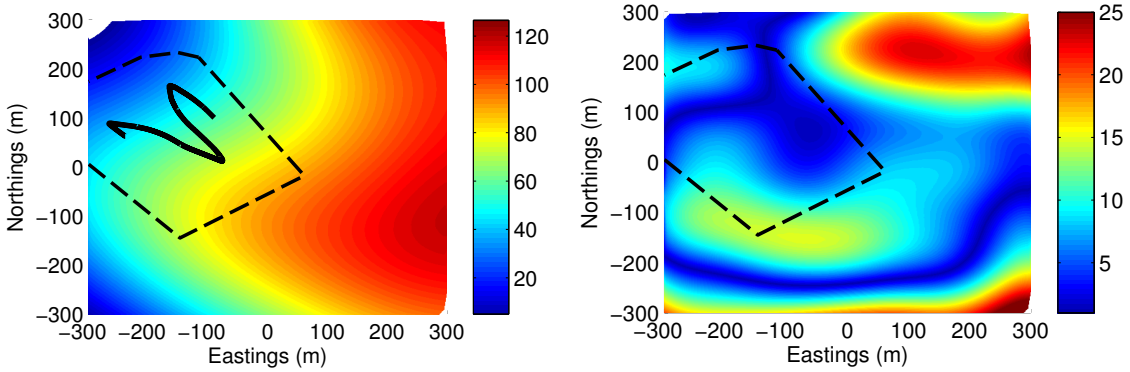
(f) GP Landmark and Slope Error

Figure 55: Terrain reconstructions using data from the first 75% of the traverse. The rover's path is shown as a solid black line, while the convex hull of landmark points is indicated by a dashed line.



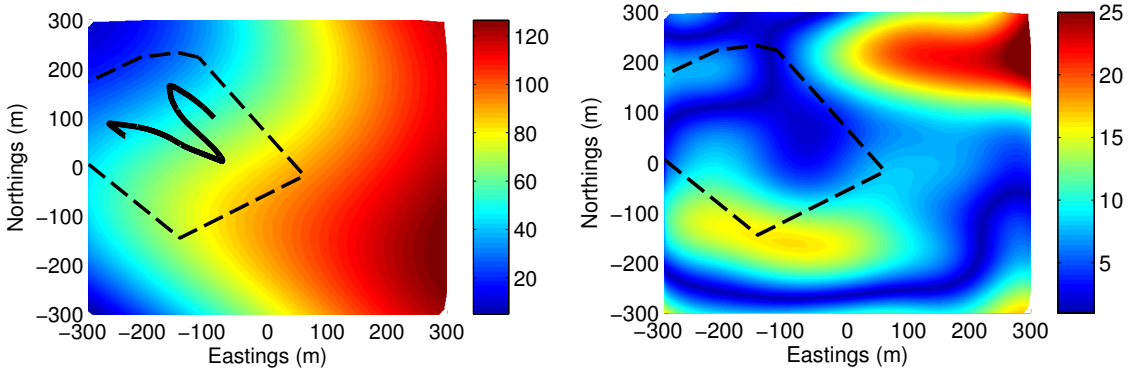
(a) Triangular Mesh Reconstructed

(b) Triangular Mesh Error



(c) GP Landmark Reconstructed

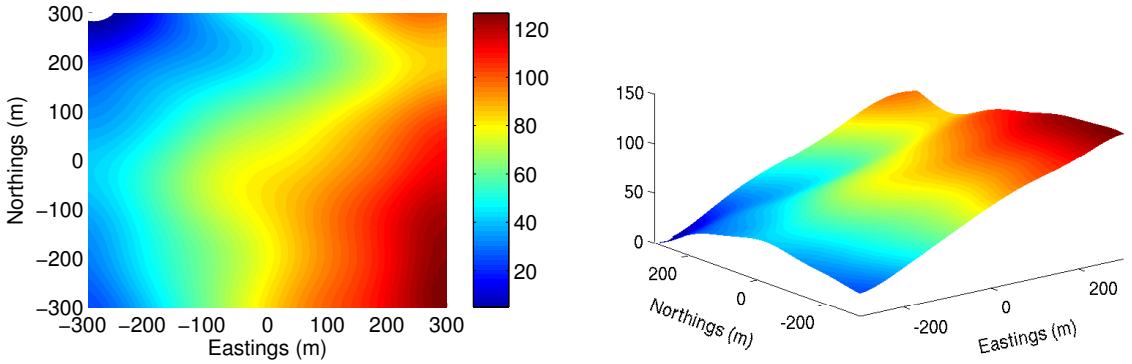
(d) GP Landmark Error



(e) GP Landmark and Slope Reconstructed

(f) GP Landmark and Slope Error

Figure 56: Terrain reconstructions using data from the complete traverse. The rover's path is shown as a solid black line, while the convex hull of landmark points is indicated by a dashed line.



(a) (b)

Figure 57: The ground truth terrain elevation from the simulated glacial environment.

hull of input points, all methods return similar RMS reconstruction errors in the range of $3m$ to $5m$. While the triangular mesh method generally returns the lowest error within the convex hull region, the reconstructed terrain is not smooth. This is particularly noticeable in the elevation error plot in Figure 56b where hard edges and error peaks are present. As the triangular mesh reconstruction is not valid outside the convex hull region, comparison is impossible. However, the GP methods are able to achieve RMS reconstruction errors below $10.0m$, with absolute errors below $25.0m$ in terrain reconstructed over $300m$ from the rover's traverse. The inclusion of the slope information reduces the reconstruction error, both in the local convex hull region, and at large distances. This improvement is most notable when the number of landmarks is small. At 25% of the data, the reconstruction using landmarks and slope reduces the RMS error by over 20% compared to the landmark-only reconstruction. However, as the number of landmarks increases, and the reconstruction approaches the true terrain, this benefit diminishes.

5.4 Conclusions

A method for performing a terrain reconstruction has been presented that leverages the previous work in terrain characterization and visual localization. This method uses a statistical construct known as a Gaussian Process, popular in the geostatistics

Table 7: Reconstruction Accuracy

Reconstruction Type	Percent Data	RMS Error (Convex Hull)	RMS Error (Entire Terrain)
Trimesh	25%	4.0 m	N/A
	50%	3.3 m	N/A
	75%	4.3 m	N/A
	100%	4.7 m	N/A
Landmarks Only	25%	3.3 m	17.5 m
	50%	3.1 m	14.2 m
	75%	7.6 m	13.6 m
	100%	6.1 m	9.8 m
Landmarks and Slope	25%	2.8 m	13.5 m
	50%	3.0 m	12.5 m
	75%	5.9 m	11.3 m
	100%	5.4 m	9.4 m

community for elevation modeling, that incorporates measurements and measurement error in a statistically sound way. However, unlike in geostatistical modeling, the input measurements utilized in this work have not been collected in an ordered manner designed to properly characterize terrain variations, but has rather been collected opportunistically, as a byproduct of the vision-based SLAM localization system. Despite this limitation, it has been shown that a high quality terrain model can still be generated, and that the behavior of terrain hundreds of meters from the robots path may be predicted accurately. When compared with standard triangular mesh interpolation of the same visual landmarks, the Gaussian Process produces similar average errors in a local region. Unlike the triangular mesh method though, the GP interpolation is not limited to a local region, and terrain elevation estimates for any location may be generated. Additionally, it has been shown that the inclusion of terrain slope information in the interpolation process yields terrain models that estimate distant terrain structure much sooner in the robot's traverse. If the purpose of creating the terrain model is to provide a map to a planning algorithm, the ability

to properly characterize the terrain earlier in the traverse is a distinct advantage. While the Gaussian Process is capable of handling these derivative measurements in a natural way, most other interpretation schemes cannot.

Additionally, as the Gaussian Process is a statistical construct, not only is the maximally likely terrain surface available, but an estimate of the reconstruction uncertainty is also produced. For planning purposes, a path could be generated that not only minimizes the threat of known hazards in the well-characterized local region, but also minimizes potential hazards by preferentially selecting paths that stay within a bounded uncertainty range. Conversely, if the goal of the mission is to construct a terrain model of some minimum quality, then knowledge of the current reconstruction uncertainty can actually drive the path planning decisions.

However, from a computational standpoint, the GP methods are at a disadvantage. Because a full measure covariance matrix must be generated, this method has a time and space complexity that is quadratic in the number of measurements used. This can be mitigated in two ways. First, only a subset of the measurements can be used, or multiple measurements can be aggregated to limit the number of measurements used. Both of these strategies have been employed in this work, where only distant landmarks are used in the reconstruction, and the large volume of slope information is averaged around an evenly distributed grid. Secondly, instead of using all of the measurements to reconstruct each query point, only the subset of measurements local to the query point need to be considered. Since the magnitude of the entries in the covariance matrix are affected only by distance between them, an effective approximation radius can be formed. However, this was found to be of marginal benefit in this example due to the large length scale used to model the glacial terrain. In this case, the entire simulated terrain fell within a reasonable approximation radius.

CHAPTER VI

CONCLUSIONS

A multi-agent robotic network has been proposed to collect weather-related sensor data from glacial regions in Greenland and Antarctica. Data collected from these remote regions will assist climate scientists in the development of more accurate climate-change models. For such a network to be successful in harsh arctic conditions, the system must exhibit high levels of fault tolerance. For multi-agent networks, fault tolerance is generally guaranteed through the use of redundant robotic nodes. Given the expendable nature of a robotic agent in this context, each node must be inexpensive, making due with low-accuracy consumer-grade sensing equipment, rather than the bleeding-edge technology a single-unit system may incorporate. Thus, GPS data alone provides insufficient accuracy when performing satellite validation tasks or collecting survey data for a terrain reconstruction; additional sources of information must be used to augment the natural capabilities of the on-board sensors.

Vision has been selected as the secondary sensing modality for both internal state estimation and terrain-awareness for the robotic agents. It is among the cheapest of available sensors, uses very little power compared to active lidar sensors, and offers a large amount of information about the environment, both of the near-field and distant regions. However, due to the snow-covered surface and low-light atmospheric conditions of arctic environments, camera images often suffer from poor contrast and lack distinctive features. Special care must be exercised when processing these types of images to preserve the limited information available.

This work has explored processes and techniques for utilizing a standard monocular camera stream to aid the navigational system of an arctic robotic sensor node.

The problem of autonomous glacial navigation has been attacked on three different fronts: hazard detection, localization, and map building or terrain reconstruction. The key results and conclusions are summarized in the following.

6.1 Key Results

6.1.1 Hazard Detection

In the area of hazard detection, a large body of research revolves around applications for Mars or military desert navigation. As such, this work focuses on the detection of discrete hazards, such as rocks or small brush. In glacial environments however, hazards tend to be local regions with a large terrain slope. Whether the rover is approaching the mouth of a crevasse, nearing a glacial lake basin, or advancing up the mountain edge, at some point the terrain incline will exceed the capabilities of the robotic platform. Hence, assessing the terrain slope is key to determining terrain traversability with glacial regions. Two related methods were presented in Chapter 3, as well as in [128, 129], that estimate the terrain slope using only a single camera image as input. These methods, inspired by heuristic knowledge used by human observers, exploit small scale surface texture that visually aligns with the perceived terrain grade. However, due to the low-contrast nature of glacial scenery, nonlinear contrast enhancement techniques must be applied before the surface texture may be extracted. These methods have been evaluated qualitatively on image sets acquired during during field trials to three different glacier systems in Alaska. Additionally, these methods were tested within a visually faithful 3D robot simulation system of one Mendenhall Glacier test site. Analysis showed a strong correlation between the visual slope estimates and the true slopes extracted from the simulation. Up to 80% of the simulated terrain slope variance could be predicted from the visual slope estimates, even though the simulated surface texture contained no directional features.

To demonstrate the effectiveness of the terrain slope data as an input to a robotic

system, a reactive control behavior was developed with the purpose of minimizing the roll experienced by the rover. A distributed behavior-based architecture was used to combine the disparate desires to drive towards the goal location and to minimize the chassis roll. This relatively simple control scheme reduced the average roll experience by the rover by over 30 percent, with the results obtained using the slope estimates nearly identical to those obtained using true slope information extracted from the simulation. However, the total rover path increased significantly with the roll-avoidance controller enabled, with no intuitive way of trading roll performance for a shorter path.

Before any meaningful analysis of terrain obstacles can be performed, the foreground region must first be segmented from the acquired image. This serves to focus subsequent processing on a smaller, targeted region of interest, as well as to remove unwanted dynamic features, such as clouds, from the image. However, in glacial images, the boundary between the traversable foreground and background elements is often subtle or ambiguous. Distant snow-covered mountains visually blend with the snow-covered foreground terrain, while low clouds and overcast skies blur the boundary between ground and sky. Several typical segmentation algorithms were applied to these images, all of which resulted in a large amount of misclassified terrain because only information local to each pixel was used during the classification. Instead, a novel horizon line detection process had been proposed in Chapter 3.1 and [130] that has been tailored to work on glacial images containing ambiguous horizons. This method searches for strong horizon segments globally, then weights each candidate using a set of heuristics to identify the best horizon segments. These segments are then connected to form the final horizon estimate. To test the classification performance, the proposed method and several common segmentation algorithms were applied to hand-labeled images from ten different field trial locations. The proposed horizon line extraction procedure consistently performed the best of all methods tested, while

requiring an order of magnitude less processing time than the second best method.

The proposed image processing techniques produce terrain slope estimates using only the current frame as input. Consequently these estimates tend to be noisy, even for a stationary camera. As there will be a high amount of visual correlation between consecutive camera frames, it should be possible to achieve better, more stable slope estimates by incorporating past measurements into an iterative estimation scheme. Additionally, while the example reactive controller demonstrated the efficacy of the slope estimates as a navigational control input, tuning the behavior weights to affect the system's slope avoidance threshold was not straightforward. Instead, the slope estimates could be projected into a global frame and converted into a standard traversability map. This would allow the application of existing traversability planning and control to be directly applied to the glacial navigation problem.

6.1.2 Localization

For the localization system, a custom vision-based SLAM system was implemented. When developing a visual odometry system for glacial images, feature extraction is possibly the biggest challenge. A procedure for extracting reliable features from these low-contrast environments has been described in Chapter 4.2 and in [131]. To validate this procedure, the quantity and consistency of extracted features in the raw image and enhanced image are compared. The preprocessing steps resulted in a five to ten fold increase in the number of detected features and the number of matched features when applied to sample image sequences from the glacial field trials. Perhaps most importantly, every enhanced image produced a minimum of ten matched features, whereas the unenhanced images often resulted in zero correctly matched features. Further, because the proposed preprocessing steps are also a requirement for the hazard detection, no additional computation time is incurred.

Although visual SLAM systems perform better than standard wheel odometry,

particularly in conditions where wheel slippage is likely, SLAM is still an incremental localization procedure. Errors accumulate over time, causing the final state estimate to drift. As the length of the expected traverse increases, the viability of using a vision-only solution diminishes. In this work, an external source of global information in the form of low-accuracy GPS was fused with the incremental localization of visual SLAM. This produced a system that is capable of both consistent local-scale localization and drift-free global-scale positioning. The full implementation details of this system are described in Chapter 4.3, with some early results published in [133]. In particular, the one-way nature to the proposed robotic mission allowed for aggressive pruning of the active landmark database, resulting in a nearly-constant active database size regardless of the total length of the traverse. This results in a nearly constant-time SLAM update that is capable of running at multiple frames per second over very long traverses.

The most problematic aspect of the visual SLAM system as implemented revolves around weighting each robot pose particle and subsequent pose sampling. The ultimate weight of each particle in the filter is the product of all of the observation probabilities. As each particle maintains an active database of on the order of 500 landmarks, even a small perturbation in pose between particles can lead to huge weighting difference. One solution is to add additional sensing capabilities to the rover platform. Such things as track odometry and even low-quality IMU measurements should result in a large improvement in the open-loop pose estimate of the rover. That would reduce the space of poses over which the particle filter would have to sample. Alternatively, different solution approaches exist. Recently introduced incremental smoothing and mapping algorithms [58, 59] are able to find the optimal solution given all of the available data. If such systems are capable of operating in near real-time, an improvement in localization quality should be possible.

6.1.3 Terrain Reconstruction

While the visual SLAM system produces a set of 3D point estimates that lie on the terrain surface as a byproduct of the localization process, this type of map is not readily utilized by common path planning algorithms. A procedure has been proposed that leverages common terrain modeling techniques from the geostatistics community. A Gaussian process (GP) terrain model has been constructed that combines information from both the sparse visual landmarks produced by the localization system and the dense slope estimates from the hazard detection algorithm. The GP framework has many advantages over simple triangular mesh interpolation. First, it is able to naturally incorporate the available dense slope information into the terrain model, improving the reconstruction accuracy at great distances. Second, it is able to predict the terrain elevation at any desired location, even in areas well beyond the currently observed data. During simulation trials, the GP terrain reconstruction was accurate within 20m of the true elevation at query points 300m from the camera, even though the visual landmark estimates do not extend beyond 75m from the camera. In contrast, triangular mesh interpolation simply cannot predict values outside the convex hull of landmark points. From a path planning perspective, the GP reconstruction allows hazardous slopes and obstacles to be accounted for much earlier in the traverse.

Unfortunately, the GP calculation is $O(n^3)$ in the number of measurements. This creates a large computational load as the length of the traverse increases, and thus the number of localized visual landmarks increases. If the dense visual slope estimates are included as well, the number of measurements increases very quickly. The current implementation takes tens of minutes to perform the final reconstruction, though the codebase has not been optimized in any way. Methods for performing approximate GP interpolation exist, often using only those measurements within a certain radius

of the current query point. Since the magnitude of the GP covariance function decreases with increasing distance between query points, a distance threshold can be created that only ignores the covariance values below a user-definable level. With coding optimizations and approximation schemes implemented, a terrain reconstruction should be possible on low-power computing hardware, but real-time operation is not to be expected.

6.2 Future Work

This research was performed as part of a larger project to produce a multi-agent reconfigurable mobile glacial weather station network. The autonomous navigation algorithms described herein are simply a small part of the overall project goal. The next logical step in the progression of this research is to design and field a larger prototype platform capable of a multiple day deployment. The current prototype platforms were designed with a three day field trial cycle in mind. For use in extended operations, several key mechanical and electrical systems need to be redesigned. Specifically, the power subsystem needs dramatic improvements, both in power management and absolute power capacity. Further, the on-board computing must upgraded if the proposed algorithms are to be tested autonomously. All time trials within this document were performed on a 2Ghz laptop, which has a significant computational advantage over the current embedded microprocessor.

Finally, the algorithms developed in this thesis have been implemented independently of each other. At a minimum, the goal-pursuit behavior used in Chapter 3.3 should be replaced by a global path planner operating on the terrain reconstruction of Chapter 5. More realistically, a supervisory system should be constructed that is responsible for switching between obstacle avoidance strategies and advanced goal pursuit strategies, as well as determining when the current map is stale and should be reconstructed from new data.

REFERENCES

[1] Agrawal, M., Konolige, K., Bolles, R., and Standford, C., "Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach," in *Proceedings of the IEEE Workshop on Applications of Computer Vision, WACV*, (Austin, Texas), Feb 2007.

[2] Akenine-Moller, T., Haines, E., and Hoffman, N., *Real-time rendering*. AK Peters Ltd, 3rd ed., 2008.

[3] Akers, E. L., Harmon, H. P., Stansbury, R. S., and Agah, A., "Design, fabrication, and evaluation of a mobile robot for polar environments," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS*, vol. 1, (Anchorage, Alaska), Sept 2004.

[4] Anderson, B., Moore, J., and Barratt, J., *Optimal filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[5] Apostolopoulos, D., Wagner, M. D., Shamah, B., Pedersen, L., Shillcutt, K., and Whittaker, W. R. L., "Technology and field demonstration of robotic search for antarctic meteorites," *International Journal of Robotics Research*, vol. 19, p. 1015, Dec 2000.

[6] Bailey, T. and Durrant-Whyte, H., "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[7] Baten, S., Dickmanns, E., Forces, A., and Burt, P., "Techniques for autonomous, off-road navigation," *IEEE Intelligent Systems Magazine*, 1998.

[8] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[9] Bekey, G., *Autonomous robots: from biological inspiration to implementation and control*. The MIT Press, 2005.

[10] Broggi, A., Bertozzi, M., Fascioli, A., Bianco, C. G. L., and Piazzi, A., "Visual perception of obstacles and vehicles for platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 164–176, 2000.

[11] Broggi, A. and Fascioli, A., "Artificial vision in extreme environments for snowcat tracks detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, pp. 162–172, Sept 2002.

[12] Christensen, H., Kirkeby, N., Kristensen, S., Knudsen, L., and Granum, E., "Model-driven vision for in-door navigation," *Robotics and Autonomous Systems*, vol. 12, no. 3, pp. 199–208, 1993.

[13] Clemente, L., Davison, A., Reid, I., Neira, J., and Tardós, J., "Mapping large loops with a single hand-held camera," in *Proceedings of Robotics: Science and Systems, RSS*, vol. 3, (Atlanta, Georgia), pp. 38–45, June 2007.

[14] Coll, C., Caselles, V., Galve, J., Valor, E., Niclos, R., Sanchez, J., and Rivas, R., "Ground measurements for the validation of land surface temperatures derived from AATSR and MODIS data," *Remote Sensing of Environment*, vol. 97, no. 3, pp. 288–300, 2005.

[15] Connor, C., Korzen, N., Knuth, E., Sauer, D., and Heavner, M., "Changing lake bathymetry with deglaciation: The mendenhall glacier system," *AGU Fall Meeting Abstracts*, Dec 2008.

[16] Cozman, F. and Krotkov, E., "Automatic mountain detection and pose estimation for teleoperation of lunar rovers," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Albuquerque, New Mexico), pp. 2452–2457, April 1997.

[17] Cressie, N., "The origins of kriging," *Mathematical Geology*, vol. 22, no. 3, pp. 239–252, 1990.

[18] Curry, J., Schramm, J., and Ebert, E., "Sea ice-albedo climate feedback mechanism," *Journal of Climate*, vol. 8, no. 2, pp. 240–247, 1995.

[19] Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., and Bradski, G., "Selfsupervised monocular road detection in desert terrain," in *Proceedings of Robotics: Science and Systems, RSS*, p. 33, The MIT Press, 2006.

[20] d'Angelo, P. and others, "Hugin - panorama photo stitcher." Online. URL: http://hugin.sourceforge.net/ [cited 9 April 2010].

[21] Davies, E., *Machine vision: theory, algorithms, practicalities*, vol. 3. Academic press London, 1990.

[22] Davies, E., *Machine vision*. Morgan Kaufmann, 3rd ed., 2005.

[23] Davis, C. H. and Segura, D. M., "An algorithm for time series analysis of ice sheet surface elevations from satellite altimetry," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 39, pp. 202–206, 2001.

[24] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1052–1067, 2007.

[25] De Berg, M., Cheong, O., Van Kreveld, M., and Overmars, M., *Computational geometry: algorithms and applications*, ch. 9, pp. 191–218. New York, NY: Springer-Verlag Inc., 2008.

[26] Deyle, T., "MIT's DARPA urban grand challenge," Nov 2008. http://www.hizook.com/blog/2008/11/04/mits-darpa-urban-grand-challenge.

[27] Deyle, T., "Hokuyo UTM-30LX laser rangefinder (LIDAR)," Feb 2009. http://www.hizook.com/projects/hokuyo-utm-30lx-laser-rangefinder-lidar.

[28] Deyle, T., "SICK LMS 291 laser rangefinder (LIDAR)," Feb 2009. http://www.hizook.com/projects/sick-lms-291-laser-rangefinder-lidar.

[29] Dijkstra, E., "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[30] Duda, R. and Hart, P. E., "Use of the hough transformation to detect lines and curves in pictures," *Communications of the Association of Computing Machinery*, vol. 15, pp. 11–15, 1972.

[31] Duda, R., Hart, P., and Stork, D., *Pattern classification*. Wiley-Interscience, 2nd ed., 2000.

[32] Durrant-Whyte, H. and Bailey, T., "Simultaneous localization and mapping (SLAM): part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[33] Eade, E. and Drummond, T., "Scalable monocular SLAM," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, (New York, NY), pp. 469–476, June 2006.

[34] Eade, E. and Drummond, T., "Unified loop closing and recovery for real time monocular SLAM," in *Proceedings of the British Machine Vision Conference*, (Leeds, UK), Sept 2008.

[35] Elfes, A., "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[36] Farr, T. and others, "The shuttle radar topography mission," *Reviews of Geophysics*, vol. 45, no. 2, 2007.

[37] Fefilatyev, S., Smarodzinava, V., Hall, L. O., and Goldgof, D. B., "Horizon detection using machine learning techniques," in *Proceedings of the 5th International Conference on Machine Learning and Applications, ICMLA*, pp. 17–21, Dec 2006.

[38] Fleming, M., Chapin, F., Cramer, W., Hufford, G., and Serreze, M., "Geographic patterns and dynamics of alaskan climate interpolated from a sparse station record," *Global Change Biology*, vol. 6, no. S1, pp. 49–58, 2000.

[39] Foessel, A., Chheda, S., and Apostolopoulos, D., "Short-range millimeter-wave radar perception in a polar environment," in *Proceedings of the Field and Service Robotics Conference, FSR*, (Pittsburgh, PA), Aug 1999.

[40] Foley, J., *Computer graphics: principles and practice*, ch. 15, pp. 651–714. Addison-Wesley Professional, 1996.

[41] Forsyth, D. A., "Shape from texture without boundaries," in *Proceedings of the European Conference on Computer Vision, ECCV*, (Copenhagen, Denmark), pp. 225–239, May 2002.

[42] Foundation, B., "Blenderwiki doc:manual." Online. URL: http://wiki.blender.org/index.php/Doc:Manual [cited 1 March 2009].

[43] Fuller, W. A., *Measurement Error Models*, ch. 4, pp. 292–360. Hoboken, NJ: John Wiley & Sons, Inc., 1987.

[44] Gerkey, B. P., Vaughan, R. T., and Howard, A., "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics, ICAR*, (Coimbra, Portugal), pp. 317–323, July 2003.

[45] geun Kim, Y. and Kim, H., "Layered ground floor detection for vision-based mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, pp. 13–18, 26 2004.

[46] Gifford, C. M., Akers, E. L., Stansbury, R. S., and Agah, A., *Mobile Robots for Polar Remote Sensing*, pp. 1–22. Springer Publishing Company, Incorporated, 2009.

[47] Goldberg, S., Maimone, M., Matthies, L., Syst, I., and Northridge, C., "Stereo vision and rover navigation software for planetary exploration," in *Proceedings of the IEEE Aerospace Conference*, vol. 5, (Big Sky, MT), March 2002.

[48] Hambrey, M., *Glacial environments*. Univ of British Columbia Pr, 1994.

[49] Harris, C. and Stephens, M., "A combined corner and edge detector," in *Proceedings of the Alvey vision conference*, (Manchester, UK), pp. 147–151, 1988.

[50] Hart, P., Nilsson, N., and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[51] Hartley, R. and Zisserman, A., *Multiview Geometry in Computer Vision*. Cambridge University Press, 2000.

[52] Hershberger, J. and Snoeyink, J., "Speeding up the douglas-peucker line simplification algorithm," in *Proceedings of the International Symposium on Spatial Data Handling*, vol. 1, (Charleston, South Carolina), pp. 134–143, Aug 1992.

[53] Hong, L., Wan, Y., and Jain, A., "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[54] Howard, A. M. and Seraji, H., "Vision-based terrain characterization and traversability assessment," *Journal of Robotic Systems*, vol. 18, no. 10, pp. 577–587, 2001.

[55] Hrabar, S., "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (Nice, France), pp. 807–814, IEEE, Sept 2008.

[56] ichi Kanatani, K. and Chou, T.-C., "Shape from texture: General principle," *Artificial Intelligence*, vol. 38, no. 1, pp. 1–48, 1989.

[57] "Juneau icefield research program," 2006. http://www.juneauicefield.com/.

[58] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F., "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Shanghai, China), p. In Press, May 2011.

[59] Kaess, M., Ranganathan, A., and Dellaert, F., "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[60] Kawagoe, M. and Tojo, A., "Fingerprint pattern classification," *Pattern Recognition*, vol. 17, no. 3, pp. 295–303, 1984.

[61] Kim, D., Oh, S., and Rehg, J., "Traversability classification for UGV navigation: A comparison of patch and superpixel representations," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (San Diego, CA), Oct 2007.

[62] Lacroix, S., Mallet, A., Bonnafous, D., Bauzil, G., Fleury, S., Herrb, M., and Chatila, R., "Autonomous rover navigation on unknown terrains: Functions and integration," *The International Journal of Robotics Research*, vol. 21, no. 10-11, p. 917, 2002.

[63] Latombe, J., *Robot motion planning*. Springer Verlag, 1990.

[64] LaValle, S., "Rapidly-exploring random trees: A new tool for path planning," tech. rep., TR 98-11, Computer Science Dept., Iowa State University, 1998.

[65] Lazzara, M. and others, "Antarctic meteorological research center (AMRC) and automatic weather station (AWS) project," Sept 2010. http://amrc.ssec.wisc.edu/.

[66] Leal, J., Scheding, S., and Dissanayake, G., "3D mapping: A stochastic approach," in *Proceedings of the Australian Conference on Robotics and Automation*, (Sydney, Australia), pp. 135–140, Nov 2001.

[67] Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., and others, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[68] Lever, J., Streeter, A., and Ray, L. R., "Performance of a solar-powered robot for polar instrument networks," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Orlando, Florida), pp. 4252–4257, May 2006.

[69] Li, H., Lavin, M. A., and Master, R. J. L., "Fast hough transform: A hierarchical approach," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2-3, pp. 139–161, 1986.

[70] Liu, J., Chen, R., and Logvinenko, T., "A theoretical framework for sequential importance sampling and resampling," *Sequential Monte Carlo Methods in Practice*, pp. 225–246, 2001.

[71] Liu, L., Crowe, T., Roberge, M., and Bakambu, J., "Vision-based exploration algorithms for rough terrain modeling using triangular mesh maps," in *Proceedings of the International Workshop on Robotic and Sensors Environments, ROSE*, (Ottawa, Ontario), pp. 1–6, IEEE, Oct 2007.

[72] "Long-term ecological research - niwot ridge," 2007. http://www.lternet.edu/sites/nwt/.

[73] Lowe, D., "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, vol. 2, pp. 1150–1157, 1999.

[74] Manduchi, R., Castano, A., Talukder, A., and Matthies, L., "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous Robots*, vol. 18, no. 1, pp. 81–102, 2005.

[75] Manz, M., Himmelsbach, M., Luettel, T., and Wuensche, H., "Fusing lidar and vision for autonomous dirt road following," *Autonome Mobile Systeme 2009*, pp. 17–24, 2009.

[76] Matas, J., Chum, O., Urban, M., and Pajdla, T., "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[77] McGill, R., Tukey, J., and Larsen, W., "Variations of box plots," *American Statistician*, pp. 12–16, 1978.

[78] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, pp. 43–72, Nov 2005.

[79] Montemerlo, M., *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, June 2003.

[80] Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D., and Strohband, S., "Winning the DARPA grand challenge with an AI robot," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, (Boston, MA), p. 982, July 2006.

[81] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 18, (Acapulco, Mexico), pp. 1151–1156, Aug 2003.

[82] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, AAAI/IAAI*, (Alberta, Canada), pp. 593–598, July 2002.

[83] Montiel, J., Civera, J., and Davison, A., "Unified inverse depth parametrization for monocular SLAM," in *Proceedings of Robotics: Science and Systems, RSS*, (Philadelphia, USA), August 2006.

[84] Morisette, J., Privette, J., Justice, C., and Running, S., "MODIS land validation plan," tech. rep., EOS Validation Program, Sept 1998.

[85] Murphy, R., "Trial by fire," *IEEE Robotics & Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.

[86] Muscato, G., Caltabiano, D., Guccione, S., Longo, D., Coltelli, M., Cristaldi, A., Pecora, E., Sacco, V., Sim, P., Virk, G., and others, "ROBOVOLC: a robot for volcano exploration result of first test campaign," *Industrial Robot: An International Journal*, vol. 30, no. 3, pp. 231–242, 2003.

[87] Naval Jr, P., "Robot self-localization from single mountain images," in *Proceedings of the Philippine Computing Science Congress*, (Quezon City, Philippines), pp. 83–88, Feb 2000.

[88] Nister, D., "An efficient solution to the Five-Point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 192–202, 2003.

[89] Nister, D., Naroditsky, O., and Bergen, J., "Visual odometry," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 1, (Washington, DC), pp. 652–659, June 2004.

[90] Nock, R. and Nielsen, F., "Statistical region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1452–1458, 2004.

[91] Pears, N. and Liang, B., "Ground plane segmentation for mobile robot visual navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, vol. 3, pp. 1513–1518, 2001.

[92] Pedersen, L., Wettergreen, D., Apostolopoulos, D., McKay, C., DiGoia, M., Heys, S., Teza, J., Wagner, M., and Ali, K. A., "Rover design for polar astrobiological exploration," in *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS*, 2005.

[93] Pupilli, M. and Calway, A., "Real-time visual SLAM with resilience to erratic motion," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 1, (New York, NY), June 2006.

[94] Rasmussen, C., "Gaussian processes in machine learning," *Advanced Lectures on Machine Learning*, pp. 63–71, 2004.

[95] Rasmussen, C. and Williams, C., "Gaussian processes for machine learning," 2006.

[96] Ray, L., Price, A., Streeter, A., Denton, D., and Lever, J. H., "The design of a mobile robot for instrument network deployment in antarctica," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Barcelona, Spain), pp. 2111–2116, April 2005.

[97] Rekleitis, I., Bedwani, J., Gingras, D., and Dupuis, E., "Experimental results for over-the-horizon planetary exploration using a lidar sensor," in *Proceedings of Experimental Robotics: The Eleventh International Symposium*, pp. 65–77, Springer Verlag, 2009.

[98] Reza, A. M., "Realization of the contrast limited adaptive histogram equalization (CLAHE) for Real-Time image enhancement," *The Journal of VLSI Signal Processing*, vol. 38, no. 1, pp. 35–44, 2004.

[99] Rosenblatt, J., "DAMN: A distributed architecture for mobile navigation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2, pp. 339–360, 1997.

[100] Saxena, A., Chung, S., and Ng, A., "Learning depth from single monocular images," *Advances in Neural Information Processing Systems*, vol. 18, p. 1161, 2006.

[101] Saxena, A., Sun, M., and Ng, A., "Make3D: Learning 3D scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 824–840, 2009.

[102] Serreze, M. and Barry, R., *The Arctic climate system*. Cambridge Univ Press, 2005.

[103] Sigel, D. and Wettergreen, D., "Star tracker celestial localization system for a lunar rover," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (San Diego, California), pp. 2851–2856, Oct 2007.

[104] Sim, R. and Little, J., "Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, (Beijing, China), pp. 2082–2089, IEEE, Oct 2006.

[105] Smith, P., Tamppari, L., Arvidson, R., Bass, D., Blaney, D., Boynton, W., Carswell, A., Catling, D., Clark, B., Duck, T., and others, "H2O at the phoenix landing site," *Science*, vol. 325, no. 5936, p. 58, 2009.

[106] Solak, E., Murray-Smith, R., Leithead, W., Leith, D., and Rasmussen, C., "Derivative observations in gaussian process models of dynamic systems," *Advances in Neural Information Processing Systems*, pp. 1057–1064, 2003.

[107] Sparbert, J., Dietmayer, K., and Streller, D., "Lane detection and street type classification using laser range images," *IEEE Proceedings in Intelligent transportation Systems*, pp. 456–464, 2001.

[108] Spikes, V. and Hamilton, G., "GLAS calibration-validation sites established on the west antarctic ice sheet.," in *Proceedings of the International Archives on Remote Sensing of the Environment*, (Honolulu, HI), Nov 2003.

[109] Squyres, S., Grotzinger, J., Arvidson, R., Bell, J., Calvin, W., Christensen, P., Clark, B., Crisp, J., Farrand, W., Herkenhoff, K., and others, "In situ evidence for an ancient aqueous environment at meridiani planum, mars," *Science*, vol. 306, no. 5702, pp. 1709–1714, 2004.

[110] Steffen, K., "Greenland climate network (GC-Net)," Dec 2010. http://cires.colorado.edu/science/groups/steffen/gcnet/.

[111] Steffen, K., Box, J., Cullen, N., and Huff, R., "Greenland climate network (GC-NET)," in *Proceedings of the Automatic Weather Stations on Glaciers*, (Pontresian, Switzerland), pp. 100–102, March 2004.

[112] Stein, M., *Interpolation of Spatial Data: some theory for kriging*. Springer Verlag, 1999.

[113] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., and others, "Stanley: The robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, pp. 661–692, 2007.

[114] Ulrich, I. and Nourbakhsh, I., "Appearance-based obstacle detection with monocular color vision," in *Proceedings of the National Conference on Artificial Intelligence*, (Austin, TX), pp. 866–871, Aug 2000.

[115] Urmson, C., Anhalt, J., Clark, M., Galatali, T., Gonzalez, J., Gowdy, J., Gutierrez, A., Harbaugh, S., Johnson-Roberson, M., Kato, H., and others, "High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004," *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-04-37*, 2004.

[116] Urmson, C., Ragusa, C., Ray, D., Anhalt, J., Bartz, D., Galatali, T., Gutierrez, A., Johnston, J., Harbaugh, S., and others, "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.

[117] Vandapel, N., Moorehead, S., Whittaker, W. R. L., Chatila, R., and Murrieta-Cid, R., "Preliminary results on the use of stereo, color cameras and laser sensors in antarctica," in *Proceedings of the International Symposium on Experimental Robotics*, (Sydney, Australia), March 1999.

[118] Vasudevan, S., Ramos, F., Nettleton, E., and Durrant-Whyte, H., "Gaussian process modeling of large-scale terrain," *Journal of Field Robotics*, vol. 26, no. 10, pp. 812–840, 2009.

[119] Vasudevan, S., Ramos, F., Nettleton, E., and Durrant-Whyte, H., "Heteroscedastic gaussian processes for data fusion in large scale terrain modeling," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, pp. 3452–3459, IEEE, 2010.

[120] Vincze, M., Schlemmer, M., Gemeiner, P., and Ayromlou, M., "Vision for robotics: A tool for model-based object tracking," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 53–64, 2005.

[121] Volpe, R., "Mars rover navigation results using sun sensor heading determination," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, vol. 1, (Kyongju, Korea), pp. 460–467, Oct 1999.

[122] Von Mises, R., *Mathematical Theory of Probability and Statistics*, ch. 9.3, pp. 427–430. New York, NY: Academic Press Inc., 1964.

[123] Wang, J., Haining, R., and Cao, Z., "Sample surveying to estimate the mean of a heterogeneous surface: reducing the error variance through zoning," *International Journal of Geographical Information Science*, vol. 24, no. 4, pp. 523–543, 2010.

[124] Wang, K., Zhou, X., Liu, J., and Sparrow, M., "Estimating surface solar radiation over complex terrain using moderate-resolution satellite sensor data," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 47–58, 2005.

[125] Wang, W., Liang, S., and Meyers, T., "Validating MODIS land surface temperature products using long-term nighttime ground measurements," *Remote Sensing of Environment*, vol. 112, no. 3, pp. 623–635, 2008.

[126] White, R. and Forsyth, D., "Combining cues: Shape from shading and texture," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 2, (New York, NY), June 2006.

[127] Williams, S., Brooks, D., and Howard, A. M., "Robot vision for science-driven navigation in challenging arctic environments," in *Robot Vision: New Research* (Matsuda, T., ed.), ch. 9, pp. 225–240, NovaPublisher, 2009.

[128] Williams, S. and Howard, A. M., "A single camera terrain slope estimation technique for natural arctic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Pasadena, CA), pp. 2729–2734, May 2008.

[129] Williams, S. and Howard, A. M., "Towards visual arctic terrain assessment," in *Proceedings of the International Conference on Field and Service Robotics, FSR*, (Cambridge, MA), July 2009.

[130] Williams, S. and Howard, A. M., "Horizon line estimation in glacial environments using multiple visual cues," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, (Shanghai, China), p. In Press, May 2011.

[131] Williams, S. and Howard, A., "Developing monocular visual pose estimation for arctic environments," *Journal of Field Robotics*, vol. 27, no. 2, pp. 145–157, 2010.

[132] Williams, S., Hurst, M., and Howard, A., "Development of a mobile arctic sensor node for earth-science data collection applications," in *Proceedings of the American Institute of Aeronautics and Astronautics, Infotech@Aerospace*, (Atlanta, GA), April 2010.

[133] Williams, S., Parker, L., and Howard, A., "Calibration and validation of earth-observing sensors using deployable surface-based sensor networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, no. 4, pp. 427–432, 2010.

[134] Williams, S., Remy, S., and Howard, A., "3D simulations for testing and validating robotic-driven applications for exploring lunar poles," in *Proceedings of the American Institute of Aeronautics and Astronautics, Infotech@Aerospace*, (Atlanta, GA), April 2010.

[135] Williams, S., Viguria, A., and Howard, A. M., "A robotic mobile sensor network for achieving scientific measurements in challenging environments," in *Proceedings of the Earth Science Technology Conference*, (Washington, DC), June 2008.

[136] Williamson, T. and Thorpe, C., "A specialized multibaseline stereo technique for obstacle detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, (Santa Barbara, CA), pp. 238–244, June 1998.

[137] Woo, J., Son, K., Li, T., Kim, G., and Kweon, I., "Vision-based UAV navigation in mountain area," in *Proceedings of the IAPR Conference on Machine Vision Applications, Japan*, (Tokyo, Japan), pp. 236–239, May 2007.

[138] Yacoob, Y. and Davis, L., "Segmentation using meta-texture saliency," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, (Rio de Janeiro, Brazil), pp. 1–8, Oct 2007.

[139] Yoerger, D., Bradley, A., Walden, B., Cormier, M., and Ryan, W., "Fine-scale seafloor survey in rugged deep-ocean terrain with an autonomous robot," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 2, (San Francisco, CA), pp. 1787–1792, April 2000.

[140] Zhang, J. and Nagel, H.-H., "Texture-based segmentation of road images," in *Proceedings of the Intelligent Vehicles Symposium*, (Paris, France), pp. 260–265, Oct 1994.

[141] Zhang, R., Tsai, P., Cryer, J., and Shah, M., "Shape-from-shading: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[142] Zwally, H., Schutz, B., Abdalati, W., and others, "ICESat's laser measurements of polar ice, atmosphere, ocean, and land," *Journal of Geodynamics*, vol. 34, no. 3-4, pp. 405–445, 2002.